

# **EXPLORER 85**

**MONITOR SOURCE LISTINGS**

**-Terminal Version-**

**COPYRIGHT © 1979**

**BY NETRONICS RESEARCH AND DEVELOPMENT LTD.**

**RT. 202 NEW MILFORD, CT 06776**

**ALL RIGHTS RESERVED**

NO PART OF THIS PUBLICATION MAY BE REPRODUCED, TRANSMITTED, TRANSCRIBED,  
STORED IN A RETRIEVAL SYSTEM, OR TRANSLATED INTO ANY LANGUAGE OR COMPUTER  
LANGUAGE, IN ANY FORM OR BY ANY MEANS, ELECTRONIC, MECHANICAL, MAGNETIC,  
OPTICAL, CHEMICAL, MANUAL OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION  
OF NETRONICS RESEARCH & DEVELOPMENT, 333 LITCHFIELD RD., NEW MILFORD, CT  
U.S.A.



**NETRONICS RESEARCH AND DEVELOPMENT LIMITED**  
**333 LITCHFIELD ROAD (RTE. 202) NEW MILFORD, CONNECTICUT 06776**  
**(203) 354-9375**

ISIS-II 8080/8085 MACRO ASSEMBLER, V2.030  
ROTOLYU ТРУБЧИТИ ГАРТ :

EXPL85 PAGE 1

4203  
328801 PRINT

LOC	OBJ	SEQ	SOURCE STATEMENT
		1	NAME EXPL85
		2 ;	
		3 ;	
		4 ;	NAME DSC030
		5 ;	DATER
		6 ;	
		7 ;*****	*****
		8 ;	ROTOLYU
		9 ;	EVASR
		10 ;	MONITOR MACROS
		11 ;*****	*****
		12 ;	STAR
		13 TRUE	MACRO WHERE ;1 BRANCH IF FUNCTION RETURNS TR
		UE	UE
		14	JC WHERE
		15 ENDM	VASR
		16 ;	
		17 FALSE	MACRO WHERE ;1 BRANCH IF FUNCTION RETURNS FA
		LSE	LSE
		18	JNC WHERE
		19 ENDM	VASR
		20 ;	
		21 ;	
		22 ;*****	*****
		23 ;	RESTART
		24 ;	RESET
		25 ;	RESTART
		26 ;	RESTART
		27 ;	RESTART
F000		28	ORG OF000H
		29 ;	
		30	BOOT:
F000	C303FO	31	JMP START ; CLEAR BOOT AND BEGIN MONITOR
		32	START:
F003	3E8D	33	MVI A, 8DH
F005	D3F2U	34	OUT VASR
F007	3E09	35	MVI A, 09H
F009	32F8FB	36	STA USCSR
F00C	D3F8	37	OUT CSR
F00E	AF	38	XRA A
F00F	32F9F8	39	STA LCOUNT
F012	D3FO	40	OUT PORTA
F014	21E5F4	41	LXI H, COUT
F017	22FDFA	42	SHLD DOUT
			; INITIALIZE OUTPUT CHANNEL TO
			CONSOLE
F01A	218AF4	43	LXI H, CIN
F01D	22FBF8	44	SHLD DIN
F020	C3BFF0	45	JMP START1
		46 ;	

LOC	OBJ	SEQ	SOURCE STATEMENT
		47 ;	
F024		48 ORG OF024H	
F024 C338F2		49 JMP STP25 ; TRAP INTERRUPT VECTOR	
		50 ;	
		51 ;	
F02C		52 ORG OF02CH	
F02C C3A6F8		53 JMP RST55	
		54 ;	
		55 ;	
F034		56 ORG OF034H	
F034 C3A3F8		57 JMP RST65	
		58 ;	
		59 ;	
F03C		60 ORG OF03CH	
F03C C34BFO		61 JMP RSAVE ; RST 7.5 VECTOR	
		62 START1:	
F03F 3E3F		63 MVI A,63 ; LOAD DISPLAY LENGTH INTO A	
F041 32FAF8		64 STA LLIMIT ; SET DISPLAY LENGTH	
F044 C335F4		65 JMP SIOTST ; GET BAUD RATE	
		66 SIORET:	
F047 FB		67 EI	
F048 C372F0		68 JMP MESAGE	
		69 RSAVE:	
F04B 22E2F8		70 SHLD LSAV ; SAVE H&L REGISTERS	
F04E E1		71 POP H ; GET USER PROG COUNTER FROM ST	
		ACK ACK	
F04F 21DCF8		72 LXI H,MNSTK	
F052 F9		73 SPHL	
F053 22E5F8		74 SHLD PSAV ; AND STORE IT	
F056 F5		75 PUSH PSW	
F057 E1		76 POP H	
F058 22EOF8		77 SHLD FSAV ; STORE FLAGS AND ACCUMULATOR	
F05B 210000		78 LXI H,O ; CLEAR H&L	
F05E 39		79 DAD SP ; GET USER STACK POINTER	
F05F 22E7F8		80 SHLD SSAV ; STORE USER STACK POINTER	
F062 21EOF8		81 LXI H,BSAV+1 ; SET STACK POINTER FOR	
		STORING THE	
F065 F9		82 SPHL ; REMAINING REGISTERS	
F066 C5		83 PUSH B ; SAVE B&C REGISTERS	
F067 D5		84 PUSH D ; SAVE D&E REGISTERS	
F068 20		85 RIM ; GET USER INTERRUPT STATUS AND	
		MASK	
F069 E60F		86 ANI OFH ; MASK UNNECESSARY BITS (----XX XX)	
F06B 32E4F8		87 STA ISAV ; SAVE INTERRUPT STATUS	
F06E 3E0E		88 MVI A,UNMSK ; UNMASK INTERRUPTS FOR MONITOR	
		USE	
F070 30		89 SIM	
F071 FB		90 EI	
		91 ;	
		92 ;	
		93 ;*****MONITOR EQUATES	
		94 ;	
		95 ;	

LOC	OBJ	SEQ	SOURCE STATEMENT	ORG	COD
		96 ;			
		97 ;*****			
		*****			
		98 ;			
0009	00	99 BITSI	EQU	9	; NUMBER OF INPUT BITS
000B	00	100 BITSO	EQU	11	; NUMBER OF OUTPUT BITS
001B	00	101 BRCHR	EQU	1BH	; CODE FOR BREAK CHARACTER (ESC APE)
F7FA		102 BRTAB	EQU	OF7FAH	; START OF BRANCH TABLE IN ROM
000D		103 CR	EQU	0DH	; CODE FOR CARRIAGE RETURN
00F8	*****	104 CSR	EQU	OF8H	; 8155 COMMAND/STATUS REGISTER ADDRESS
001B	00	105 ESC	EQU	1BH	; CODE FOR ESCAPE CHARACTER
000F		106 HCHAR	EQU	OFH	; MASK FOR LOWER HALF OF BYTE
00FF		107 INVRT	EQU	OFFH	; MASK TO INVERT HALF BYTE FLAG
000A		108 LF	EQU	0AH	; CODE FOR LINE FEED
0000		109 LOWER	EQU	0	; LOWER HALF BYTE IN ICMD
F8DC		110 MNSTK	EQU	OF8DC	; START OF MONITOR STACK
00F0		111 PORTA	EQU	OFOH	; SYSTEM PORT A
00F2		112 PORTAD	EQU	OF2H	; PORT A DIRECTION REGISTER
00F1		113 PORTB	EQU	OF1H	; PORT B
00F3		114 PORTBD	EQU	OF3H	; PORT B DIRECTION REGISTER
00FB		115 PORTC	EQU	OFBH	; 8155 PORT C
00F9		116 PROUT	EQU	OF9H	; PARALLEL OUTPUT PORT (8155 PORT A)
00FA	00	117 PRIN	EQU	OF8H	; PARALLEL INPUT PORT (8155 PORT B)
007F		118 PRTYD	EQU	07FH	; MASK TO CLEAR PARITY FROM CONSOLE CHAR
F8A6		119 RST55	EQU	OF8A6H	
F8A3		120 RST65	EQU	OF8A3H	
F800	00	121 RAMST	EQU	OF800H	; START OF RAM
0022		122 RMUSE	EQU	34	; MONITOR RAM USAGE
0080		123 SSTRT	EQU	80H	; SHIFTED START BIT
00C0		124 STRT	EQU	0C0H	; UNSHIFTED START BIT
0040		125 STOFB	EQU	40H	; STOP BIT
001B		126 TERM	EQU	1BH	; CODE FOR ICMD TERMINATION (ESCAPE)
00FD		127 TIMHI	EQU	OFDH	
00FC		128 TIMLO	EQU	OFCH	
000E	00	129 UNMSK	EQU	OEH	; UNMASK INPUT INTERRUPT
00FF		130 UPPER	EQU	OFFH	; UPPER HALF BYTE IN ICMD
F8A0		131 USINT	EQU	OF8A0H	
		132 ;			
		133 ;			
		134 ;			
		135 ;			
		136 ;			
		137 ;			
		138 ;			
		139 ;			
		140 ;			
		141 ;			
		142 MESAGE:			
F072	2142F7	143 MSHRS	LXI	H, SINON	; GET ADDRESS OF SIGNON

LOC	OBJ	SEQ	SOURCE STATEMENT		
			LOC	OBJ	MESSAGE
F075	064A	144	MVI	B,LSGNON	; GET CHARACTER COUNT
		145	MSGL:		
F077	4E	146	MOV	C,M	; FETCH CHARACTER TO C REGISTER
F078	CDDAF4	147	CALL	CO	; OUTPUT CHARACTER TO CONSOLE
F07B	23	148	INX	H	; POINT TO NEXT CHARACTER
F07C	05	149	DCR	B	; DECREMENT CHARACTER COUNTER
F07D	C277FO	150	JNZ	MSGL	; RETURN FOR NEXT CHARACTER
		151	;		
		152	;		
		153	;		*****
		154	;		*
		155	;		*COMMAND RECOGNIZING ROUTINE*
		156	;		*
		157	;		*****
		158	;		
		159	; FUNCTION:	GETCM	
		160	; INPUTS:	NONE	
		161	; OUTPUTS:	NONE	
		162	; CALLS:	GETCH,ECHO,ERROR	
		163	; DESTROYS:	A,B,C,H,L,F/F'S	
		164	; DESCRIPTION:	GETCM RECIEVES AN INPUT CHARACTER FROM THE USER AND ATTEMPTS	
		165		TO LOCATE IT IN ITS COMMAND TABLE. IF SUCCESSFUL, THE ROU-	
		166		TINE CORRESPONDING TO THIS CHARACTER IS	
		167		SELECTED FROM ITS	
		168		TABLE OF COMMAND ROUTINE ADDRESSES, AND	
		169		CONTROL IS TRANS-	
		170		FERRED TO THIS ROUTINE. IF THE CHARACTE	
		171		R DOES NOT MATCH	
		172	GETCM:		
F080	AF	173	XRA	A	
F081	21DCF8	174	LXI	H,MNSTK	; RESET STACK POINTER TO MONITO
			R		
F084	F9	175	SPHL		
F085	OE2E	176	MVI	C,'.'	; PROMPT CHARACTER TO C REGISTE
			R		
F087	CD14F5	177	CALL	ECHO	; OUTPUT PROMPT
F08A	C38DF0	178	JMP	GTC03	; LEAVE ROOM FOR RST BRANCH
		179	GTC03:		
F08D	CD63F5	180	CALL	GETCH	; GET COMMAND CHARACTER TO A
F090	CD14F5	181	CALL	ECHO	; ECHO CHARACTER TO USER
F093	79	182	MOV	A,C	
F094	010B00	183	LXI	B,NCMDS	; C CONTAINS LOOP AND INDEX COU
			NT		
F097	21B2F7	184	LXI	H,CTAB	; H&L POINTS TO COMMAND TABLE
		185	GTC05:		
F09A	BE	186	CMP	M	; COMPARE ENTRY TO TABLE CHARAC
			TER		
F09B	CAA6FO	187	JZ	GTC10	; BRANCH IF EQUAL COMMAND RECO

LOC	OBJ	SEQ	SOURCE STATEMENT	OPC	REG	MEM
FO9E	OBJ1 DT \$0000	188	INX H	904		
			; ELSE, POINT TO NEXT CHARACTER			
FO9F	OD	189	DCR H	905	C	
			; DECREMENT LOOP COUNTER			
FOA0	C29AF0	190	JNZ H, GTC05	906		
			; BRANCH IF NOT AT TABLE END			
FOA3	C34CF5	191	JMP ERROR	907		
			; ELSE, COMMAND CHARACTER IS I			
			LLEGAL			
		192	GTC10:			
FOA6	219AF7	193	LXI H, CADR	908		
			; FOR GOOD COMMAND, LOAD ADDR			
FOA9	09TARABEE DT	194	DAD B	909		
			; ADD WHAT IS LEFT OF THE LOOP			
FOAA	09TXEA DT	195	DAD B	910		
			; OFFSET THE H&L POINTER TO THE			
FOAB	7E	196	MOV A, M	911		
			; GET LSB OF ADDRESS TO A			
FOAC	23	197	INX H	912		
			; INCREMENT POINTER (ADDRESS IS			
			2 BYTES LONG)			
FOAD	66	198	MOV H, M	913		
			; MOV MSB TO H REGISTER			
FOAE	6F	199	MOV L, A	914		
			; MOVE LSBYTE TO L REGISTER			
FOAF	E9	200	PCHL	915		
			; NEXT INSTRUCTION COMES FROM CMD ROUTI			
		201	NE			
		202				
		203				
		204	*****			
		205				
		206	COMMAND IMPLEMENTING ROUTINES			
		207				
		208				
		209	*****			
		210				
		211				
		212	; FUNCTION: DCMD			
		213	; INPUTS: NONE			
		214	; OUTPUTS: NONE			
		215	; CALLS: ECHO, NMOUT, HILO, GETCM, CRDUT, GETNM			
		216	; DESTROYS: A, B, C, D, E, H, L, F/F'S			
		217	; DESCRIPTION: DCMD IMPLEMENTS THE DISPLAY MEMORY COMMA			
		218	ND ("D")			
		219				
		220	DCMD:			
FOB0	0E02	221	MVI C, 2	916		
			; GET TWO NUMBERS FROM INPUT			
FOB2	CDC8F5	222	CALL GETNM	917		
			; INPUTS: A, B			
FOB5	3AFAF8	223	LDA LLIMIT	918		
			; GET CURRENT LINE LENGTH			
FOB8	FE3E	224	CPI 62	919		
			; INPUTS: A			
FOB9	DAC2F0	225	JC DCM06	920		
			; INPUTS: B			
FOB0	3EOF	226	MVI A, OFH	921		
			; INPUTS: C			
FOB1	C3C4F0	227	JMP DCM07	922		
			; INPUTS: D			
		228	DCM06:			
FOC2	3E07	229	MVI A, 07H	923		
			; INPUTS: E			
		230	DCM07:			
FOC4	32F7F8	231	STA NEWLN	924		
			; SET NEW LINE MASK			

LOC	OBJ	SEQ	SOURCE STATEMENT			
FOC7	D1	232	POP	D	; ENDING ADDRESS TO D&E	
FOC8	E1	233	POP	H	; BEGINNING ADDRESS TO H&L	
FOC9	CD06F5	234	DCM05:			
FOCC	7C	235	CALL	CROUT	; OUTPUT CR/LF TO CONSOLE	
FODD	CD45F6	236	MOV	A,H	; DISPLAY ADDRESS OF FIRST LOCA	
TION						
FODD	CD45F6	237	CALL	NMOUT		
FODD	7D	238	MOV	A,L	; ADDRESS IS 2 BYTES LONG	
FOD1	CD45F6	239	CALL	NMOUT		
FOD4	OE20	240	DCM10:			
FOD6	CD14F5	241	MVI	C,' '	; USE BLANK TO SEPARATE DATA	
FOD9	7E	242	CALL	ECHO		
FODA	CD45F6	243	MOV	A,M	; GET CONTENTS OF NEXT MEMORY L	
OCATION						
FODD	CD45F6	244	CALL	NMOUT	; DISPLAY CONTENTS	
FODD	CD0DF6	245	CALL	HILO	; CHECK FOR LAST ADDRESS OF DIS	
PLAY DATA						
FODD	CD06F5	246	FALSE	DCM15	; IF NOT, DISPLAY MORE DATA	
FOE0	D2E9F0	247+	JNC	DCM15		
FOE3	CD06F5	248	CALL	CROUT	; END WITH CR/LF	
FOE6	C380F0	249	JMP	GETCM	; DONE. GET NEXT COMMAND	
FOE9	23	250	DCM15:			
FOEA	3AF7F8	251	INX	H	; POINT TO NEXT CHARACTER	
FOED	A5	252	LDA	NEWLN		
FOEE	C2D4F0	253	ANA	L	; CHECK FOR START OF NEW LINE	
FOF1	C3C9F0	254	JNZ	DCM10	; NO - NOT A NEW LINE	
		255	JMP	DCM05	; YES - BEGIN NEW LINE	
		256				
		257				
		258	*****	*****		
		259	*****	*****		
		260	*****	*****		
		261	; FUNCTION: TCMD			
		262	; INPUTS: NONE			
		263	; OUTPUTS: NONE			
		264	; CALLS: ERROR, CROUT			
		265	; DESTROYS: A, F/F'S			
		266	; DESCRIPTION: TCMD IT USED TO MANUALLY TURN THE TAPE I/O CONTROL			
		267		CHANNEL ON AND OFF TO ALLOW FOR REWINDIN		
		268		G, ETC.		
		269		WHEN CALLED, TCMD LOOKS FOR THE NUMBER "0" TO IN-		
		270		DICATE OFF, OR THE NUMBER "1" TO INDICAT		
		271		E ON. OTHER		
		272		CHARACTERS WILL CAUSE A CALL TO THE ERRO		
		273		R HANDLER.		
		274	TCMD:			
FOF4	CD63F5	275	CALL	GETCH	; GET A CHARACTER (1 OR 0)	
FOF7	CD14F5	276	CALL	ECHO		
FOFA	79	277	MOV	A,C		

LOC	OBJ	SEQ	SOURCE STATEMENT	SEG	SEG
F0FB	FE30	278	CPI '0'		
F0FD	CA0E1	279	JZ TCM00 ; IF IS'S 0 - BRANCH		
F100	FE31	280	CPI '1'		
F102	C24CF5	281	JNZ ERROR ; IF NEITHER 1 OR 0 ERROR		
F105	DBF0	282	IN PORTA		
F107	F608	283	ORI 08H		
F109	D3F0	284	OUT PORTA	SET CTRL BIT = 1	
F10B	C314F1	285	JMP TCM01		
		286	TCM00:	RETURNCODE	
F10E	DBF0	287	IN C,B	PORTA	
F110	E6F7	288	ANI GCM01	OF7H	RETURNCODE
F112	D3F0	289	OUT PORTA	; SET CTRL BIT = 0	
		290	TCM01:		
F114	CD06F5	291	CALL CROUT		
F117	C380F0	292	JMP GETCM VM		
		293	; RETURNCODE	CALL JAC	
		294	; VM	JVM	
		295	;*****		
		296		RETURNCODE	
		297	; RATE = 0	RDS	100
		298	; FUNCTION: GCM01		
		299	; INPUTS: NONE	CAL	
		300	; OUTPUTS: A,C	VM	
		301	; CALLS: ERROR, GETHX, RSTTF		
		302	; DESTROYS: A,B,C,D,E,H,L,F/F'S		
		303	; DESCRIPTION: GCM01 IMPLEMENTS THE BEGIN EXECUTION COMM AND ("G")		
		304	; RETURNCODE		
		305	; VALID FOR AVALID ENTRY		
		306	GCM01:		
F11A	CD6AF5	307	CALL GETHXURT	GET ADDR (IF PRESENT) FROM IN	
		308	PUT		
		309	FALSE JAV GCM05	BRANCH IF NO NUMBER PRESENT	
F11D	D22FF1	310	JNC GCM05		
F120	7A00	311	MOV GCM01,A,D	ELSE, GET TERMINATOR	
F121	FE0D	312	CPI CR	SEE IF CARRIAGE RETURN	
F123	C24CF5	313	JNZ GCM05	ERROR IF NOT PROPERLY TERMINA	
		314	LXI A,H,PSAV	REPLACE PROG COUNTER W/START	
		315	ADDRESS		
F129	71	316	MOV M,C		
F12A	23	317	INX AH		
F12B	70F	318	MOV M,B		
F12C	C335F1	319	JMP GCM10		
		320	GCM05:		
F12F	7A00	321	MOV A,D IRX	WITH NO START ADDR, CHECK FOR	
		322	CRT		
F130	FE0D	323	CPI CR		
F132	C24CF5	324	JNZ ERROR	ERROR IF NOT PROPERLY TERMINA	
		325	TED		
		326	GCM10:		
		327	JMP RSTORJAC	RESTORE REGISTERS, BEGIN EXEC	
		328	RETURNCODE		
		329	FUNCTION		
		330	NEXT COMM		
		331	ENTER		
		332			

LOC	OBJ	SEQ	SOURCE STATEMENT
		326	; ****
			*****
		327	;
		328	;
		329	; FUNCTION: ICMD
		330	; INPUTS: NONE
		331	; OUTPUTS: NONE
		332	; CALLS: ERROR, ECHO, GETCH, VALDL, VALDG, CNVBN, STHLF, GETNM, CROUT
		333	; DESTROYS: A, B, C, D, E, H, L, F/F'S
		334	; DESCRIPTION: ICMD IMPLEMENTS THE INSERT CODE INTO MEM COMMAND ("I")
		335	;
		336	ICMD:
F138	OE01	337	MVI C,1
F13A	CDC8F5	338	CALL GETNM ; GET ONE NUMBER FROM INPUT
F13D	3EFF	339	MVI A, UPPER
F13F	32F4F8	340	STA TEMP ; TEMP HOLDS THE UPPER/LOWER HA
			LFBYTE FLAG
F142	D1	341	POP D ; START ADDRESS TO D&E
		342	ICM05:
F143	CD63F5	343	CALL GETCH ; GET A CHARACTER FROM INPUT
F146	4F	344	MOV C,A
F147	CD14F5	345	CALL ECHO ; ECHO CHARACTER TO CONSOLE
F14A	79	346	MOV A,C ; MOV CHARACTER BACK TO A
F14B	FE1B	347	CPI TERM ; CHECK FOR TERMINATING CHARACT
			ER
F14D	CA79F1	348	JZ ICM25 ; IF TERMINATOR, DONE ENTERING
F150	CD03F7	349	CALL VALDL ; ELSE, CHECK FOR VALID DELIMIT
			ER
F153	DA43F1	350	TRUE ICM05 ; IF DELIMITER IGNORE CHARACTER
F156	CDE8F6	351+	JC ICM05
		352	CALL VALDG ; ELSE, CHECK FOR VALID HEX CHA
			RACTER
		353	FALSE ICM20 ; IF NOT, BRANCH TO ERROR HANDL
			ER
F159	D273F1	354+	JNC ICM20
F15C	CD5AF5	355	CALL CNVBN ; CONVERT DIGIT TO BINARY
F15F	4F	356	MOV C,A
F160	CDC9F6	357	CALL STHLF ; STORE APPROPRIATE HALFWORD
F163	3AF4F8	358	LDA TEMP ; GET HALFBYTE FLAG
F166	B7	359	ORA A ; SET F/F'S
F167	C26BF1	360	JNZ ICM10 ; BRANCH IF FLAG SET FOR UPPER
F16A	13	361	INX D ; IF LOWER, INCREMENT ADDRESS
		362	ICM10:
F16B	EEFF	363	XRI INVRT ; TOGGLE STATE OF FLAG
F16D	32F4F8	364	STA TEMP ; STORE FLAG
F170	C343F1	365	JMP ICM05 ; PROCESS NEXT DIGIT
		366	ICM20:
F173	CDBEF6	367	CALL STHFO ; ILLEGAL CHARACTER
F176	C34CF5	368	JMP ERROR ; FILL ENTIRE BYTE BEFORE ERROR
F179	CDBEF6	369	ICM25: CALL STHFO ; JMP HERE FOR ESCAPE CHARACTER
F17C	CD06F5	370	CALL CROUT ; SEND OUT LF/CR
F17F	CD80F0	371	CALL GETCM ; RETURN FOR NEXT COMMAND
		372	:

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ
		373 ;			
		374 ;			
		375 ;*****			
		376 ; T30 : XH30 CALL R30			
		377 ;			
		378 ; FUNCTION: MCMD			
		379 ; INPUTS: NONE			
		380 ; OUTPUTS: NONE			
		381 ; CALLS: GETCM, HILO, GETNM			
		382 ; DESTROYS: A, B, C, D, H, L, F/F'S			
		383 ; DESCRIPTION: MCMD IMPLEMENTS THE MOVE DATA IN MEMORY COMMAND ("M")			
		384 ; ON: MCMD			
		385 MCM05:			
F182	OE03	386 MVI C,3			
F184	CDC8F5	387 CALL GETNM ; GET THREE NUMBERS FROM INPUT			
F187	C1	388 POP B ; GET DESTINATION ADDRESS TO B&C			
F188	E1	389 POP H ; GET ENDING ADDRESS TO H&L			
F189	D1	390 POP D ; GET STARTING ADDRESS TO D&E			
F18A	E5	391 MCM05:			
F18B	62	392 PUSH H ; SAVE ENDING ADDRESS			
F18C	68	393 MOV H,D ; SOURCE ADDRESS TO H&L			
F18D	7E	394 MOV L,E ; GET SOURCE BYTE			
F18E	60	395 MOV A,M ; DESTINATION ADDRESS TO H&L			
F18F	69	396 MOV H,B ; MOVE BYTE TO DESTINATION			
F190	77	397 MOV L,C ; INCREMENT DESTINATION ADDRESS			
F191	03	398 INX B ; TEST FOR DESTINATION ADDRESS			
F192	78	399 MOV A,B			
F193	B1	400 ORA C			
		OVERFLOW			
		401 ; (I.E. FROM FFFF TO 0000)			
F194	CA80FO	402 JZ GETCM ; IF SO, TERMINATE AND GET NEXT COMMAND			
F197	13	403 INX D ; INCREMENT SOURCE ADDRESS			
F198	E1	404 POP H ; GET BACK ENDING ADDRESS			
F199	C00DF6	405 CALL HILO ; CHECK FOR ENDING ADDR >= SOUR CE ADDR			
F19A	07	406 FALSE GETCM ; IF NOT, COMMAND IS DONE			
F19C	D280FO	407 JNC GETCM			
F19F	C38AF1	408+ JMP MCM05 ; GET ANOTHER BYTE			
		409			
		410 ;			
		411 ; ****			
		412 ;*****			
		413 ;			
		414 ;			
		415 ; FUNCTION: CCMD			
		416 ; INPUTS: NONE			
		417 ; OUTPUTS: NONE			
		418 ; CALLS: GETHX, GETCM, NMOUT, ECHO			
		419 ; DESTROYS: A, B, C, D, E, H, L, F/F'S			
		420 ; DESCRIPTION: CCMD IMPLEMENTS THE CHANGE DATA INTO MEM			

LOC	OBJ	SEQ	SOURCE STATEMENT			
		SEQ	DRY COMMAND ("C")			
		421	;			
		422	;			
		423	CCMD:			
F1A2	CD6AF5	424	CALL	GETHX	; GET A NUMBER, IF PRESENT, FROM M INPUT	
F1A5	C5	425	PUSH	B		
F1A6	E1	426	POP	H	; GET NUMBER TO H&L (DENOTES ME MORY ADDR)	
F1A7	7A	427	CCM05:	MOV	A,D	; GET TERMINATOR FROM INPUT
F1A8	FE20	428		CPI	' '	; SEE IF SPACE
F1AA	CAB2F1	429		JZ	CCM10	; YES - CONTINUE
F1AD	FE2C	430		CPI	','	; ELSE - SEE IF COMMA
F1AF	C280F0	431		JNZ	GETCM	; NO - TERMINATE COMMAND
		432	CCM10:			
F1B2	7E	433	MOV	A,M	; GET THE CONTENTS OF THE MEMOR Y ADDR	
F1B3	CD45F6	434	CALL	NMOUT	; DISPLAY AT CONSOLE	
F1B6	OE2D	435	MVI	C,'-'		
F1B8	CD14F5	436	CALL	ECHO	; USE "-" AS SEPARATOR	
F1BB	CD6AF5	437	CALL	GETHX	; GET NEW VALUE FOR MEMORY FROM INPUT	
		438	FALSE	CCM15	; IF NO VALUE PRESENT, BRANCH	
F1BE	D2C2F1	439+	JNC	CCM15		
F1C1	71	440	MOV	M,C	; ELSE - STORE DATA IN MEMORY	
F1C2	23	441	CCM15:			
		442	INX	H	; INCREMENT POINTER TO NEXT ADD RESS	
F1C3	C3A7F1	443	JMP	CCM05		
		444	;			
		445	;			
		446	;			
		447	*****			
		448	;			
		449	; FUNCTION:	SCMD		
		450	; INPUTS:	NONE		
		451	; OUTPUTS:	NONE		
		452	; CALLS:			
		453	; DESTROYS:	EVERYTHING		
		454	; DESCRIPTION:	SCMD IMPLEMENTS THE SINGLE STEP COMMAND. WHEN IN-		
		455	;	VOKED, EACH TIME THE SPACE BAR IS PRESSE		
		456	D, THE	USERS PROGRAM IS STEPPED ONE INSTRUCTION		
		457	;	AND THE		
		458	;	THE CURRENT	CURRENT PROGRAM COUNTER IS DISPLAYED ON	
		459	;		CONSOLE DEVICE.	
		460	;			
		461	SCMD:			
F1C6	CD6AF5	462	CALL	GETHX	; GET ADDRESS, IF PRESENT, FROM CONSOLE	
		463	FALSE	STP05	; BRANCH IF NO NUMBER PRESENT	

LOC	OBJ	SEQ	SOURCE STATEMENT	OPR	OPD
F1C9	D2DEF1	464+	JNC STP05		
F1CC	7A	465	MOV A,D ; ELSE, GET TERMINATOR		
F1CD	FE0D	466	CPI CR ; WHICH MUST BE A SPACE		
F1CF	C24CF5	467	JNZ ERROR		
F1D2	21E5F8	468	LXI H,PSAV ; CHANGE PC STORAGE TO NEW VALUE		
F1D5	71	469	MOV M,C		
F1D6	23	470	INX H		
F1D7	70	471	MOV M,B		
F1D8	C006F5	472	CALL CROUT		
F1DB	C307F2	473	JMP STP20 ; CONTINUE		
		474 STP05:			
F1DE	7A	475	MOV A,D TBL ; WITH NO START ADDRESS, LOOK FOR SPACE		
F1DF	FE0D	476	CPI CR		
F1E1	C24CF5	477	JNZ ERROR		
F1E4	C006F5	478	CALL CROUT		
F1E7	C307F2	479	JMP STP20		
		480 STP15:			
F1EA	CD34F6	481	CALL DISPC		
F1ED	CD06F5	482	CALL CROUT		
		483 STP15A:			
F1F0	CD63F5	484	CALL GETCH		
F1F3	79	485	MOV A,C		
F1F4	FE20	486	CPI /,IXJ		
F1F6	CA07F2	487	JZ STP20		
F1F9	FE0D	488	CPI CR		
F1FB	CA01F2	489	JZ STP16		
F1FE	C380F0	490	JMP GETCM		
		491 STP16:			
F201	CD68F6	492	CALL REGDS		
F204	C3F0F1	493	JMP STP15A		
		494 STP20:			
F207	3AE4F8	495	LDA ISAV ; GET USER INTERRUPT MASK		
F20A	E608	496	ANI 08H		
F20C	32F4F8	497	STA TEMP ; SAVE INTERRUPT STATUS		
F20F	2AE5F8	498	LHLD PSAV		
F212	7E	499	MOV A,M ; GET CURRENT INSTRUCTION		
F213	FEF3	500	CPI OF3H ; CHECK FOR "DI"		
F215	C21CF2	501	JNZ STP21		
F218	AF	502	XRA A		
F219	C323F2	503	JMP STP22		
		504 STP21:			
F21C	FEFB	505	CPI OFBH ; CHECK FOR "EI"		
F21E	C226F2	506	JNZ STP23		
F221	3E08	507	MVI A,08H		
		508 STP22:			
F223	32F4F8	509	STA TEMP ; SAVE NEW INTERRUPT STATUS		
		510 STP23:			
F226	3E40	511	MVI A,40H		
F228	D3FD	512	OUT TIMHI		
F22A	3EC5	513	MVI A,0C5H → 197 <sub>10</sub>		
F22C	D8FC	514	OUT TIMLO ; INIT. TIMER		
F22E	3AF8F8 (13)	515	LDA USCSCR (211/10 CLK CYCLES)		
F231	F600 (7)	516	ORI 0COH		

LOC	OBJ	SEQ	SOURCE STATEMENT		
F233	D3F8	(10)	517	OUT	CSR
F235	C39FF5	(10)	518	JMP	RSTOR ; RESTORE USER REGISTERS AND GO
			519 ;		
			520 ; THIS IS THE LOCATION BRANCHED TO BY THE TRAP VECTOR,		
			521 ; AFTER ONE INSTRUCTION HAS BEEN EXECUTED.		
			522 ;		
			523 STP25:		
F238	F5		524	PUSH	PSW
F239	3AF8F8		525	LDA	USCSR ; GET USER CSR
F23C	E63F		526	ANI	3FH
F23E	F640		527	ORI	40H
F240	D3F8		528	OUT	CSR ; STOP TIMER
F242	F1		529	POP	PSW
F243	22E2F8		530	SHLD	LSAV
F246	E1		531	POP	H
F247	22E5F8		532	SHLD	PSAV
F24A	F5		533	PUSH	PSW
F24B	E1		534	POP	H
F24C	22EOF8		535	SHLD	FSAV
F24F	210000		536	LXI	H,0
F252	39		537	DAD	SP
F253	22E7F8		538	SHLD	SSAV
F256	21EOF8		539	LXI	H,BSAV+1
F259	F9		540	SPHL	
F25A	C5		541	PUSH	B
F25B	D5		542	PUSH	D
F25C	20		543	RIM	
F25D	E607		544	ANI	07H
F25F	21F4F8		545	LXI	H,TEMP
F262	B6		546	ORA	M
F263	32E4F8		547	STA	ISAV
F266	3EOE		548	MVI	A,UNMSK
F268	30		549	SIM	
F269	C3EAF1		550	JMP	STP15
			551 ;		
			552 ;*****		
			553 ;		
			554 ; FUNCTION: XCMD		
			555 ; INPUTS: NONE		
			556 ; OUTPUTS: NONE		
			557 ; CALLS: GETCH, ECHO, REGDS, GETCM, ERROR, RGADR, NMOUT		
			, CROUT, GETHX		
			558 ; DESTROYS: A, B, C, D, H, L, F/F'S		
			559 ; DESCRIPTION: XCMD IMPLEMENTS THE REGISTER EXAMINE/CHANGE COMMAND ("X")		
			560 ;		
			561 ;		
			562 XCMD:		
F26C	CD63F5		563	CALL	GETCH ; GET REGISTER IDENTIFIER
F26F	4F		564	MOV	C,A
F270	CD14F5		565	CALL	ECHO ; ECHO IT
F273	79		566	MOV	A,C
F274	FE0D		567	CPI	CR ; CHECK FOR CARRIAGE RETURN

LOC	OBJ	SEQ	SOURCE STATEMENT	930	980	990
F276	C27CF2	568	JNZ XCM05 ; BRANCH IF NOT A CR			
F279	CD68F6	569	CALL REGDS ; ELSE - DISPLAY REGISTER CONTE NTS			
		570	XCM05:			
F27C	4F	571	MOV C,A ; GET REGISTER IDENTIFIER TO C			
F27D	CD45F6	572	CALL RGADR ; CONVERT IDENTIFIER TO A TABLE ADDRESS			
F280	C5	573	PUSH B			
F281	E1	574	POP H ; PUT POINTER INTO H&L REGISTER S			
F282	OE20	575	MVI C,' '			
F284	CD14F5	576	CALL ECHO ; OUTPUT ONE SPACE TO USER			
F287	79	577	MOV A,C			
F288	32F4F8	578	STA TEMP ; PUT SPACE IN TEMP STORAGE AS DELIMITER.			
		579	XCM10:			
F28B	3AF4F8	580	LDA TEMP ; GET TERMINATOR			
F28E	FE20	581	CPI ',' ; CHECK FOR SPACE CHARACTER			
F290	CA98F2	582	JZ XCM15 ; YES - GO TO TABLE FOR DATA			
F293	FE2C	583	CPI ',' ; NO - CHECK FOR COMMA			
F295	C280F0	584	JNZ GETCM ; NO - TERMINATE ONLY ON CR			
		585	XCM15:			
F298	7E	586	MOV A,M ; GET REGISTER DATA FROM MEMORY			
F299	B7	587	ORA A ; SET F/F'S			
F29A	C2A3F2	588	JNZ XCM18 ; BRANCH IF NOT AT END OF TABLE			
F29D	CD06F5	589	CALL CROUT ; OUTPUT CR/LF TO CONSOLE			
F2A0	C380F0	590	JMP GETCM ; EXIT			
		591	XCM18:			
F2A3	E5	592	PUSH H ; PUT POINTER ON STACK			
F2A4	5E	593	MOV E,M			
F2A5	16FB	594	MVI D, RAMST SHR 8 ; ADDR OF SAVE LOC FROM TABLE			
F2A7	23	595	INX H			
F2A8	46	596	MOV B,M ; GET LENGTH FLAG FROM TABLE			
F2A9	D5	597	PUSH D ; SAVE ADDRESS OF SAVE LOCATION			
F2AA	D5	598	PUSH D			
F2AB	E1	599	POP H ; MOVE SAVE LOC INTO H&L			
F2AC	C5	600	PUSH B ; SAVE LENGTH FLAG			
F2AD	7E	601	MOV A,M ; GET REGISTER DATA FROM SAVE L OCATION			
F2AE	CD45F6	602	CALL NMOUT ; TRANSMIT TO CONSOLE			
F2B1	F1	603	POP PSW ; GET BACK LENGTH FLAG			
F2B2	F5	604	PUSH PSW ; SAVE IT AGAIN (PSW NOW SET WI TH LENGTH FLAG)			
F2B3	B7	605	ORA A ; SET F/F'S			
F2B4	CABCFF2	606	JZ XCM20 ; IF 8 BIT REG, NOTHING MORE TO DISPLAY			
F2B7	2B	607	DCX H ; IF 16 BIT REG, GET LOWER 8 BI TS			
F2B8	7E	608	MOV A,M			
F2B9	CD45F6	609	CALL NMOUT ; DISPLAY LOWER 8 BITS			
		610	XCM20:			
F2BC	OE20	611	MVI C,'-'			
F2BE	CD14F5	612	CALL ECHO ; USE DASH AS SEPARATOR			
F2C1	CD6AF5	613	CALL GETHX ; SEE IF THERE IS A NEW VALUE F			

LOC	OBJ	SEQ	SOURCE STATEMENT		
			LOC	OBJ	SEQ ROM INPUT
			614		FALSE XCM30 ; NO - CHECK FOR NEXT REGISTER
F2C4	D2DCF2	615+	JNC	XCM30	
F2C7	7A	616	MOV	A,D	
F2C8	32F4F8	617	STA	TEMP	; ELSE - SAVE THE TERMINATOR
F2CB	F1	618	POP	PSW	; GET BACK LENGTH FLAG
F2CC	E1	619	POP	H	; PUT SAVE ADDR INTO H&L
F2CD	B7	620	ORA	A	; SET F/F'S
F2CE	CAD3F2	621	JZ	XCM25	; IF 8 BIT REGISTER - BRANCH
F2D1	70	622	MOV	M,B	; SAVE UPPER 8 BITS
F2D2	2B	623	DCX	H	; POINT TO SAVE LOC FOR LOWER 8
					BITS
		624	XCM25:		
F2D3	71	625	MOV	M,C	; STORE
		626	XCM27:		
F2D4	110300	627	LXI	D,RTABS	; SIZE OF ENTRY IN RTAB TABLE
F2D7	E1	628	POP	H	; POINTER INTO REG TABLE
F2D8	19	629	DAD	D	; ADD ENTRY SIZE TO POINTER
F2D9	C38BF2	630	JMP	XCM10	; DO NEXT REGISTER
		631	XCM30:		
F2DC	7A	632	MOV	A,D	; GET TERMINATOR
F2DD	32F4F8	633	STA	TEMP	; SAVE IN MEMORY
F2EO	D1	634	POP	D	; CLEAR STACK
F2E1	D1	635	POP	D	; CLEAR STACK
F2E2	C3D4F2	636	JMP	XCM27	; GO INCREMENT REG TABLE POINTE
			R		
		637	;		
		638	*****		
		639	;		
		640	;		
		641	;		
		642	;		
		643	FUNCTION:	PCMD	
		644	INPUTS:	NONE	
		645	OUTPUTS:	NONE	
		646	CALLS:	GETNM, WAIT, HILO, BYTOUT, BITO	
		647	DESTROYS:	EVERYTHING	
		648	DESCRIPTION:	PCMD IMPLEMENTS THE STORE DATA ON MAGNET	
				IC TAPE PROGRAM	
		649		WHEN CALLED, IT EXPECTS 3 NUMBERS TO BE	
				ENTERED 1) BEGIN-	
		650		NING ADDRESS, 2) ENDING ADDRESS, AND 3)	
				PROGRAM NUMBER(0-FF)	
		651	;		
		652	;		
		653	;		
		654	RCMD:		
F2E5	OE03	655	MVI	C,3	; COUNT OF NUMBERS NEEDED
F2E7	CDC8F5	656	CALL	GETNM	; GET ADDRESSES
F2EA	C1	657	POP	B	
F2EB	79	658	MOV	A,C	; GET PROGRAM NO.
F2EC	32F5F8	659	STA	PROGNO	; AND STORE IT
F2EF	D1	660	POP	D	; GET ENDING ADDRESS
F2FO	E1	661	POP	H	" BEGINNING ADDRESS

LOC	OBJ	SEQ	SOURCE STATEMENT	REG	160	REG
F2F1	CD0DF6	662	CALL HILO ARX; CHECK FOR BAKEA			
F2F4	DA4CF5	663	JC AT030 ERRORTUO			
F2F7	3E01	664	MVI B,01H			
F2F9	D3F0	665	OUTMOT30 PORTA#1; TURN RECORDER ON			
F2FB	CD16F7	666	CALL WAIT ; 3 SECOND DELAY			
		667	*****GENERATE LEADER*****			
		668	*****			
F2FE	010070	669	LXI B,7000H ; LENGTH OF LEADER (15 SEC)			
		670	LDR1:			
F301	37	671	STC			
F302	CD69F3	672	CALL BIT0			
F305	0D	673	DCR C			
F306	C201F3	674	JNZ LDR1			
F309	05	675	DCR B			
F30A	C201F3	676	JNZ LDR1 ; DONE?			
F30D	AF	677	XRA A			
F30E	CD69F3	678	CALL BIT0 10; OUTPUT START BIT			
		679	*****RECORD RECORD NUMBER*****			
		680	*****			
F311	3AF5F8	681	LDA PROGNO ; GET RECORD NUMBER			
F314	4FCUAF3	682	MOV B,A			
F315	CD4AF3	683	CALL BYTOUT			
		684	**RECORD ENDING ADDRESS***			
		685	*****			
		686	*****			
F318	4A	687	MOV C,D ; GET UPPER BYTE			
F319	CD4AF3	688	CALL BYTOUT			
F31C	4B3	689	MOV C,E 10A; GET LOWER BYTE			
F31D	CD4AF3	690	CALL BYTOUT			
		691	*****RECORD BEGINNING ADDRESS*****			
		692	*****			
F320	4C	693	MOV C,H			
F321	CD4AF3	694	CALL BYTOUT			
F324	4D	695	MOV C,L			
F325	CD4AF3	696	CALL BYTOUT			
		697	*****RECORD DATA*****			
		698	*****			
		699	*****			
F328	4E	700	DATA: MOV C,M ; GET DATA FROM MEMORY			
F329	CD4AF3	701	CALL BYTOUT ; OUTPUT IT			
F32C	23	702	INX H			
F32D	CD0DF6	703	CALL HILO			
F330	D228F3	704	JNC DATA			
F333	4E	705	MOV C,M ; GET LAST BYTE			
F334	CD4AF3	706	CALL BYTOUT			
		707	*****			
		708	*****			
		709	*****OUTPUT TRAILER*****			
		710	*****			
F337	0664	711	MVI B,100			
		712	TRAIL: XRA A			
F339	AF	713	CALL BIT0			
F33A	CD69F3	714	DCR B			
F33D	05	715	JNZ TRAIL			
F33E	C239F3	716				

LOC	OBJ	SEQ	SOURCE STATEMENT
F341	AF	717	XRA A
F342	D3FO	718	OUT PORTA ; TURN RECORDER OFF
F344	CD06F5	719	CALL CROUT
F347	C380FO	720	JMP PATH GETCM
		721 ;	
		722 ;*****	*****
		723 ;	
		724 ;	
		725 ;	
		726 ; FUNCTION: BYTOUT	
		727 ; DESCRIPTION: THIS ROUTINE OUTPUTS ONE BYTE OF DATA TO	
		THE RECORDING	
		728 ; DEVICE, AND A PARITY BIT.	
		729 ;	
		730 BYTOUT:	
F34A	F3	731	DI
F34B	D5	732	PUSH D
F34C	0608	733	MVI B,8 ; # OF BITS TO BE OUTPUT
		734 BYT01:	
F34E	79	735	MOV A,C ; GET DATA TO THE ACCUMULATOR
F34F	07	736	RLC ; BIT TO BE OUTPUT GOES IN CY
F350	4F	737	MOV C,A ; PUT IT BACK
F351	CD69F3	738	CALL BITO
F354	05	739	DCR B
F355	C24EF3	740	JNZ BYT01 ; CONTINUE
F358	79	741	MOV A,C
F359	C600	742	ADI O ; DONE TO SET PARITY FLAG
F35B	EA62F3	743	JPE BYT02
F35E	37	744	STC
F35F	C363F3	745	JMP BYT03
		746 BYT02:	
F362	AF	747	XRA A
		748 BYT03:	
F363	CD69F3	749	CALL BITO ; OUTPUT PARITY BIT
F366	D1	750	POP D
F367	FB	751	EI
F368	C9	752	RET
		753 ;	
		754 ; FUNCTION: BITO	
		755 ; DESCRIPTION: WHEN CALLED, BITO OUTPUTS A PULSE TO THE	
		RECORDING	
		756 ; DEVICE. THE WIDTH OF THE PULSE IS DETER	
		MINED BY THE	
		757 ; CONDITION OF THE CARRY FLAG.	
		758 ;	
		759 ;	
		760 BITO:	
F369	E5	761	PUSH H
F36A	3EC0	762	MVI A,OCOH
F36C	30	763	SIM ; SET SOD TO "1"
F36D	DA82F3	764	JC BIT05
F370	2E7E	765	MVI L,126 ; IF CY=1, OUTPUT 126COUNT PULS
		E	
		766 BIT10:	

LOC	OBJ	SEQ	SOURCE STATEMENT	632	680	681
F372	2D	767	DCR 0084 L ATB	418	882300 8P63	
F373	C272F3	768	JNZ 01A0 BIT10	718	186100 8A63	
F376	3E40	769	MVI A,40H	618		
F378	30	770	SIM 0084 IVM	718		
F379	2E7E	771	MVI L,126	718		
		772	BIT15:	718		
F37B	2D	773	DCR 0118 L ATB	718	882300 8A63	
F37C	C27BF3	774	JNZ 0103 BIT15	608	883AA0 8A63	
F37F	C391F3	775	JMP 0103 BIT30L	108	884A80 8A63	
		776	BIT05:	718		
F382	2E32	777	MVI L,50 ; IF CY=0, OUTPUT 50 COUNT PULS			
		E		418	882300 8A63	
		778	BIT20:	718		
F384	2D	779	DCR 0118 L ATB	608	882300 8A63	
F385	C284F3	780	JNZ 0103 BIT20	608	883AA0 8A63	
F388	3E40	781	MVI A,40H	608	884A80 8A63	
F38A	30	782	SIM 0084 IVM	708	882300 8A63	
F38B	2E32	783	MVI L,50	608	882300 8A63	
		784	BIT25:	708	882300 8A63	
F38D	2D	785	DCR 0118 L ATB	608	882300 8A63	
F38E	C28DF3	786	JNZ 0103 BIT25	608	883AA0 8A63	
		787	BIT30:	708	882300 8A63	
F391	E1	788	POP H VQM	608	882300 8A63	
F392	C9	789	RET	608	716 8A63	
		790	;	608	882300 8A63	
		791	;	608	882300 8A63	
		792	;	608	882300 8A63	
		793	*****	608	882300 8A63	
		794	;	608	882300 8A63	
		795	;	608	882300 8A63	
		796	; FUNCTION: LCMD	608	882300 8A63	
		797	; INPUTS: NONE	608	882300 8A63	
		798	; OUTPUTS: NONE	608	882300 8A63	
		799	; CALLS: GETNM, WAIT, BITIN, BYTIN, HILO, TERROR, SRET, FRET	608	882300 8A63	
		800	; DESCRIPTION: LCMD IMPLEMENTS THE LOAD DATA FROM MAGNETIC TAPE	608	882300 8A63	
		801	; PROGRAM. WHEN CALLED, IT EXPECTS TO SEE A PROGRAM	608	882300 8A63	
		802	; A NUMBER (0-FF), THEN TURNS THE TAPE DEVICE ON AND SEARCHES	608	882300 8A63	
		803	; FOR THE PROGRAM WITH A CORRESPONDING NUMBER. WHEN FOUND	608	882300 8A63	
		804	THE DATA IS LOADED INTO THE APPROPRIATE MEMORY LOCATIONS.	608	882300 8A63	
		805	;	608	882300 8A63	
		806	;	608	882300 8A63	
		807	LCMD:	608	882300 8A63	
F393	0E01	808	MVI H10 C,1	608		
F395	CDC8F5	809	CALL 0118 GETNM ; GET PROGRAM NUMBER TO STACK			
F398	3E01	810	MVI A,01H			
F39A	D3F0	811	OUT PORTA			
F39C	C1	812	POP B			
F39D	79	813	MOV A,C			
			; STORE PROGRAM NUMBER			

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ
F39E	32F5F8	814	STA PROGNO		
F3A1	CD16F7	815	CALL WAIT		
F3A4	0620	816	LCMD1:		
		817	MVI B,20H ; 32 "1"s MUST BE DETECTED(LEAD ER)		
		818	LCMD2:		
F3A6	CDE9F3	819	CALL BITIN		
F3A9	DAAFF3	820	JC LCMD3		
F3AC	C3A4F3	821	JMP LCMD1 ; START OVER IF NOT A 1		
		822	LCMD3:		
F3AF	05	823	DCR B		
F3B0	C2A6F3	824	JNZ LCMD2 ; CONTINUE		
		825	LCMD4:		
F3B3	CDE9F3	826	CALL BITIN		
F3B6	DAB3F3	827	JC LCMD4 ; WAIT FOR A ZERO		
F3B9	21F5F8	828	LXI H,PROGNO		
F3BC	CD07F4	829	CALL BYTIN ; READ PROGNO FROM TAPE		
F3BF	BE	830	CMP M ; RIGHT ONE?		
F3C0	C2A4F3	831	JNZ LCMD1 ; NO-LOOK FOR NEXT LEADER!		
F3C3	CD07F4	832	CALL BYTIN ; YES-GET STARTING ADDR. & LENG		
		TH			
F3C6	57	833	MOV D,A ; MSB OF LENGTH		
F3C7	CD07F4	834	CALL BYTIN		
F3CA	5F	835	MOV E,A ; LSB OF LENGTH		
F3CB	CD07F4	836	CALL BYTIN		
F3CE	67	837	MOV H,A ; MSB OF BEGINNING ADDRESS		
F3CF	CD07F4	838	CALL BYTIN		
F3D2	6F	839	MOV L,A ; LSB OF BEGINNING ADDRESS		
		840	LCMD6:		
F3D3	CD07F4	841	CALL BYTIN		
F3D6	77	842	MOV M,A		
F3D7	23	843	INX H ; POINT TO NEXT ADDRESS		
F3D8	CD0DF6	844	CALL HILO		
F3DB	D2D3F3	845	JNC LCMD6 ; CONTINUE		
F3DE	CD07F4	846	CALL BYTIN ; GET THE LAST BYTE		
F3E1	77	847	MOV M,A		
		848	;		
		849	;		
		850	; ALL DONE		
		851	;		
F3E2	3E40	852	MVI A,40H		
F3E4	D3FO	853	OUT PORTA ; TURN RECORDER OFF		
F3E6	C380FO	854	JMP GETCM ; GET NEXT COMMAND		
		855	;		
		856	;		
		857	*****		
		858	;		
		859	;		
		860	; FUNCTION: BITIN		
		861	; DESCRIPTION: BITIN LOOKS FOR ONE BIT OF INFORMATION F ROM THE TAPE		
		862	; INPUT DEVICE AND RETURNS THE CARRY FLAG SET AC CORDINGLY		
		863	;		

LOC	OBJ	SEQ	SOURCE STATEMENT	DISP	OPCODE	OPNAME
		864	;			
		865	BITIN:			
F3E9 D5		866	PUSH D			
		867	BITIN1:			
F3EA 20		868	RIM			
F3EB 17		869	RAL			
F3EC D2EAF3		870	JNC	BITIN1 ; WAIT FOR ONE		
F3EF 1E2A		871	MVI	E,42		
		872	BITIN2:			
F3F1 1D		873	DCR	E		
F3F2 CAFE93		874	JZ	BITIN4		
F3F5 20		875	RIM			
F3F6 17		876	RAL			
F3F7 DAF1F3		877	JC	BITIN2		
F3FA D1		878	POP	D		
F3FB C3BCF6		879	JMP	SRET		
		880	BITIN4:			
F3FE 20		881	RIM			
F3FF 17		882	RAL			
F400 DAEF93		883	JC	BITIN4 ; WAIT FOR A ZERO BEFORE RETURN		
		884	POP	D		
F403 D1		885	JMP	FRET		
F404 C357F5		886	;			
		887	;			
		888	; FUNCTION: BYTIN			
		889	; DESCRIPTION: BYTIN RETURNS ONE BYTE OF INFORMATION FR			
		890	OM THE			
		891	MAGNETIC TAPE DEVICE. THE DATA IS CHECK			
		892	ED FOR PARITY			
		893	ERRORS BEFORE RETURNING.			
		894	;			
		894	BYTIN:			
F407 C5		895	PUSH	B		
F408 D5		896	PUSH	D		
F409 010000		897	LXI	B,0 ; CLEAR B&C		
F40C 1608		898	MVI	D,B ; LOAD CHARACTER COUNT		
		899	BYTO1:			
F40E CDE9F3		900	CALL	BITIN		
F411 D215F4		901	JNC	BYTO2 ; IF DATA=0 DON'T INCREMENT B		
F414 04		902	INR	B		
		903	BYTO2:			
F415 79		904	MOV	A,C		
F416 17		905	RAL			
F417 4F		906	MOV	C,A		
F418 15		907	DCR	D		
F419 C20EF4		908	JNZ	BYTO1		
F41C 78		909	MOV	A,B		
F41D E601		910	ANI	01H		
F41F CA2BF4		911	JZ	BYTO3 ; PARITY MUST BE A 1		
F422 CDE9F3		912	CALL	BITIN		
F425 D299F6		913	JNC	TERROR		
F428 C331F4		914	JMP	BYTO4		
		915	BYTO3:			

LOC	OBJ	SEQ	SOURCE STATEMENT	
F42B	CDE9F3	916	CALL	BITIN
F42E	DA99F6	917	JC	TERROR
		918	BYTO4:	
F431	79	919	MOV	A,C
F432	D1	920	POP	D
F433	C1	921	POP	B
F434	C9	922	RET	
		923	;	
		924	*****	*****
		925	;	
		926	;	
		927	;	UTILITY ROUTINES
		928	;	
		929	;	
		930	;	
		931	; FUNCTION	SIOTST
		932	; INPUTS:	NONE
		933	; OUTPUTS:	BAUD RATE CONSTANTS (BITTIME AND HALFBIT )
		934	; CALLS:	NOTHING
		935	; DESTROYS:	EVERYTHING
		936	; DESCRIPTION:	WHEN CALLED THE SYSTEM WAITS FOR THE "SPACE" BAR
		937	;	ON THE CONSOLE INPUT DEVICE TO BE PRESSED D, THE
		938	;	SERIAL DATA STREAM IS THEN ANALYSED TO DETER-
		939	;	MINE THE BAUD RATE AND THE APPROPRIATE TIME CON-
		940	;	stants are stored in the system memory.
		941	;	
		942	;	
		943	SIOTST:	
F435	21DCF8	944	LXI	H,MNSTK
F438	F9	945	SPHL	
F439	3EC0	946	MVI	A,0COH
F43B	30	947	SIM	; SET SERIAL OUT TO LOGIC 1
		948	BRID:	
F43C	20	949	RIM	
F43D	B7	950	ORA	A
F43E	F23CF4	951	JP	BRID ; WAIT UNTIL LOGIC 1 IS SEEN ON SERIAL IN LINE
		952	BRI1:	
F441	20	953	RIM	; GET SERIAL INPUT STATUS
F442	B7	954	ORA	A
F443	FA41F4	955	JM	BRI1 ; WAIT FOR START BIT TO BE RECEIVED
		956	EVED	
F446	21FAFF	956	LXI	H,-6 ; INITIALIZE A COUNT IN H&L
F449	1E04	957	BRI3:	
		958	MVI	E,04H
		959	BRI4:	
F44B	1D	960	DCR	E ; GENERATE A 53 MACHINE CYCLE L OOP
F44C	C24BF4	961	JNZ	BRI4

LOC	OBJ	SEQ	SOURCE STATEMENT	REG	REG
F44F	23	962	INX H	R001	C001
F450	20	963	RIM	R001	C001
F451	B7	964	ORA A	R001	C001
F452	F249F4	965	JP BRI3		
		966 ;	H&L NOW CORRESPONDS TO INCOMMING BAUD RATE		
		967 ;			
F455	E5	968	PUSH H	R001	C001
F456	24	969	INR H	R001	C001
F457	2C	970	INR L	R001	C001
F458	22E9F8	971	SHLD BITTIME		
F45B	E1	972	POP H	R001	C001
F45C	B7	973	ORA A	R001	C001
F45D	7C	974	MOV A,H	R001	C001
F45E	1F	975	RAR	R001	C001
F45F	67	976	MOV H,A	R001	C001
F460	7D	977	MOV A,L	R001	C001
F461	1F	978	RAR	R001	C001
F462	6F	979	MOV L,A	R001	C001
F463	24	980	INR H	R001	C001
F464	2C	981	INR L	R001	C001
F465	22EBF8	982	SHLD HALFBIT	R001	C001
F468	C32CF7	983	JMP LIMITST	R001	C001
		984 ;			
		985 ;			
		986 ;*****			
		987 ;			
		988 ; FUNCTION: FCMD			
		989 ; INPUTS: NONE			
		990 ; OUTPUTS: NONE			
		991 ; CALLS: GETNM,HILO,ERROR			
		992 ; DESTROYS: A,B,C,D,H,L,F/F'S			
		993 ; DESCRIPTION: FCMD IMPLEMENTS THE FILL MEMORY WITH CONSTANT COMMAND			
		994 ;			
		995 ;			
		996 FCMD:			
F46B	0E03	997	MVI C,3	R001	C001
F46D	CDC8F5	998	CALL GETNM	R001	C001
F470	C1	999	POP B	R001	C001
		1000 POP D	R001	C001	
F471	D1	1001 POP H	R001	C001	
F472	E1	1002 MOV M,C	R001	C001	
F473	71	1003 FCM05:			
F474	23	1004 INX H			
		1005 FCM10: TO BE FILLED			
F475	71	1006 MOV M,C			
F476	CD0DF6	1007 CALL HILO			
F479	D274F4	1008 JNC FCM05			
		N			

LOC	OBJ	SEQ	SOURCE STATEMENT
F47C	C380F0	1009	JMP GETCM ; ELSE - END, GET NEXT COMMAND
		1010 ;	
		1011 ;	
		1012 ;*****	*****
		1013 ;	
		1014 ;	
		1015 ; FUNCTION: CI	
		1016 ; INPUTS: NONE	
		1017 ; OUTPUTS: A - CHARACTER FROM CONSOLE	
		1018 ; CALLS: DELAY	
		1019 ; DESTROYS: A,F/F'S	
		1020 ; DESCRIPTION: WHEN CALLED, CI WAITS UNTIL A CHARACTER HAS BEEN	
		1021 ; ENTERED AT THE CONSOLE AND RETURNS THE C HARACTER,	
		1022 ; VIA THE A REGISTER, TO THE CALLING ROUTINE.	
		1023 ; MAY BE CALLED BY THE USER VIA A JUMP TAB LE IN RAM.	
		1024 ;	
		1025 ;	
		1026 CI:	
F47F	E5	1027	PUSH H
F480	F5	1028	PUSH PSW
F481	21FBF8	1029	LXI H,DIN
F484	7E	1030	MOV A,M
F485	23	1031	INX H
F486	66	1032	MOV H,M
F487	6F	1033	MOV L,A
F488	F1	1034	POP PSW
F489	E9	1035	PCHL
		1036 CIN:	
F48A	F3	1037	DI
F48B	C5	1038	PUSH B
F48C	0609	1039	MVI B,BITSI ; NUMBER OF BITS TO BE READ
		1040 CI1:	
F48E	20	1041	RIM
F48F	B7	1042	ORA A
F490	FABEF4	1043	JM CI1 ; WAIT FOR START BIT TO BEGIN
F493	2AEBF8	1044	LHLD HALFBIT
		1045 CI2:	
F496	2D	1046	DCR L
F497	C296F4	1047	JNZ CI2
F49A	25	1048	DCR H
F49B	C296F4	1049	JNZ CI2
		1050 CI3:	
F49E	2AE9F8	1051	LHLD BITTIME ; WAIT ONE BIT TIME
		1052 CI4:	
F4A1	2D	1053	DCR L
F4A2	C2A1F4	1054	JNZ CI4
F4A5	25	1055	DCR H
F4A6	C2A1F4	1056	JNZ CI4
F4A9	20	1057	RIM ; CHECK SERIAL INPUT LEVEL (1 0

LOC	OBJ	SEQ	SOURCE STATEMENT
		1058	RAL
F4AA	17	1059	DCR B ; KEEP COUNT
F4AB	05	1060	JZ CI5
F4AC	CAB6F4	1061	MOV A,C
F4AF	79	1062	RAR
F4B0	1F	1063	MOV C,A ; SAVE BIT IN C REGISTER
F4B1	4F	1064	NOP
F4B2	00	1065	JMP CI3 ; RETURN FOR NEXT BIT
F4B3	C39EF4	1066	CI5:
F4B6	79	1067	MOV A,C ; LEAVE CHARACTER IN THE A REGI
			STER
F4B7	C1	1068	POP B
F4B8	E1	1069	POP H
F4B9	FB	1070	EI
F4BA	C9	1071	RET
		1072	;
		1073	*****
		1074	;
		1075	;
		1076	; FUNCTION: PIN
		1077	; INPUTS: NONE
		1078	; OUTPUTS: A - CHARACTER FROM PARALLEL INPUT PORT
		1079	; CALLS: NOTHING
		1080	; DESTROYS: A,F/F'S
		1081	; DESCRIPTION: PIN IS A PARALLEL INPUT ROUTINE WHICH CA N BE USED IN
		1082	PLACE OF THE CONSOLE INPUT ROUTINE BY PL ACING ITS ADDRESS
		1083	IN THE INPUT CHANNEL LOCATION, "DIN".
		1084	;
		1085	;
		1086	PIN:
F4BB	DBF8	1087	IN CSR
F4BD	E610	1088	ANI 10H
F4BF	CABBF4	1089	JZ PIN ; WAIT TILL BUFFER FULL SIGNAL
F4C2	DBFA	1090	IN PRLIN ; GET THE DATA TO A
F4C4	2F	1091	CMA ; INPUT DATA MUST BE INVERTED
F4C5	C9	1092	RET
		1093	;
		1094	;
		1095	;
		1096	*****
		1097	*****
		1098	; FUNCTION: POUT
		1099	; INPUTS: C - CHARACTER TO BE OUTPUT
		1100	; OUTPUTS: C - CHARACTER OUTPUT
		1101	; CALLS: NOTHING
		1102	; DESTROYS: A,F/F'S
		1103	; DESCRIPTION: POUT IS A PARALLEL OUTPUT ROUTINE WHICH CAN BE USED IN
		1104	PLACE OF THE CONSOLE OUTPUT BY PLACING I TS ADDRESS IN
		1105	THE OUTPUT CHANNEL LOCATION, "DOUT".

LOC	OBJ	SEQ	SOURCE STATEMENT			
		1106 ;				
		1107 POUT:				
F4C6	DBF8	1108 IN CSR ;	CSR	;	GET STATUS REGISTER OF 8155	
F4C8	E602	1109 ANI 02H ;	02H	;		
F4CA	C2C6F4	1110 JNZ POUT ;	POUT	;	WAIT TILL BUFFER EMPTY	
F4CD	79	1111 MOV A,C ;	A,C	;		
F4CE	2F	1112 CMA ;		;	INVERT DATA TO COMPENSATE FOR	
		BUFFERS(74LS04)				
F4CF	D3F9	1113 OUT PRLOUT ;	PRLOUT	;	OUTPUT THE DATA	
F4D1	C9	1114 RET ;				
		1115 ;				
		1116 ;				
		1117 ;*****				
		*****				
		1118 ;				
		1119 ;				
		1120 ; FUNCTION: CO/BCO				
		1121 ; INPUTS: C - CHARACTER TO BE OUTPUT TO CONSOLE(CO)				
		)				
		1122 ;	B - " " " "	"	"	(BC
		O)				
		1123 ; OUTPUTS: C - CHARACTER OUTPUT (CO)				
		1124 ; A,B - " " (BCO)				
		1125 ; CALLS: NOTHING				
		1126 ; DESTROYS: A,F/F'S				
		1127 ; DESCRIPTION: CO SENDS ITS INPUT, THE CHARACTER IN THE				
		C REGISTER,				
		1128 ;	TO THE CONSOLE DEVICE, WHILE BCO SENDS T			
		HE CHARACTER				
		1129 ;	IN THE B REGISTER.			
		1130 ;				
		1131 ;				
		1132 ;				
		1133 BCO:				
F4D2	C5	1134 PUSH B ;	B	;	SAVE THE B&C REGISTERS	
F4D3	48	1135 MOV C,B ;	C,B	;	PUT THE OUTPUT DATA IN C	
F4D4	CDDAF4	1136 CALL CO ;	CO	;	AND OUTPUT IT	
F4D7	C1	1137 POP B ;	B	;		
F4D8	78	1138 MOV A,B ;	A,B	;	LEAVE THE CHARACTER IN A	
F4D9	C9	1139 RET ;				
		1140 ;				
		1141 ;				
		1142 CO:				
F4DA	E5	1143 PUSH H ;	H	;		
F4DB	F5	1144 PUSH PSW ;	PSW	;		
F4DC	21FDF8	1145 LXI H,DOUT ;	H,DOUT	;		
F4DF	7E	1146 MOV A,M ;	A,M	;		
F4EO	23	1147 INX H ;	H	;		
F4E1	66	1148 MOV H,M ;	H,M	;		
F4E2	6F	1149 MOV L,A ;	L,A	;		
F4E3	F1	1150 POP PSW ;	PSW	;		
F4E4	E9	1151 PCHL ;		;		
		1152 COUT:				
F4E5	F3	1153 DI ;		;		
F4E6	C5	1154 PUSH B ;	B	;	SAVE THE B&C REGISTERS	

LOC	OBJ	SEQ	SOURCE STATEMENT		
F4E7	060B	1155	MVI	B,BITSO	; SET THE NUMBER OF BITS TO BE
		OUTPUT			
F4E9	AF	1156	XRA	A	; CLEAR CARRY
		1157	C01:		
F4EA	3E80	1158	MVI	A,80H	
F4EC	1F	1159	RAR		
F4ED	3F	1160	CMC		
F4EE	30	1161	SIM		; OUTPUT DATA BIT
F4EF	2AE9F8	1162	LHLD	BITTIME	
		1163	C02:		
F4F2	2D	1164	DCR	L	; TIME OUT BAUD RATE
F4F3	C2F2F4	1165	JNZ	C02	
F4F6	25	1166	DCR	H	
F4F7	C2F2F4	1167	JNZ	C02	
F4FA	37	1168	STC		; SET UP FOR STOP BIT
F4FB	79	1169	MOV	A,C	; GET CHARACTER TO BE TRANSMITT
		ED			
F4FC	1F	1170	RAR		
F4FD	4F	1171	MOV	C,A	; STORE ROTATED DATA
F4FE	05	1172	DCR	B	
F4FF	C2EAF4	1173	JNZ	C01	; OUTPUT NEXT BIT
F502	C1	1174	POP	B	
F503	E1	1175	POP	H	; ALL DONE
F504	FB	1176	EI		
F505	C9	1177	RET		
		1178			
		1179			
		1180			
		1181			
		1182			
		1183	*****		
		1184	*****		
		1185	*****		
		1186	; FUNCTION: CROUT		
		1187	; INPUTS: NONE		
		1188	; OUTPUTS: NONE		
		1189	; CALLS: ECHO		
		1190	; DESTROYS: A,B,C,F/F'S		
		1191	; DESCRIPTION: CROUT SENDS A CARRIAGE RETURN (AND HENCE		
			A LINE FEED)		
		1192		TO THE CONSOLE.	
		1193			
		1194			
		1195	CROUT:		
F506	OEOD	1196	MVI	C,CR	; PLACE THE CARRIAGE RETURN CHA
			RACTER IN THE		
		1197			
F508	CD14F5	1198	CALL	ECHO	; C REGISTER
			T		
		1199			
F50B	C9	1200	RET		
		1201			
		1202			
		1203	*****		

LOC	OBJ	SEQ	SOURCE STATEMENT
		*****	*****
		1204 ;	
		1205 ; FUNCTION:	ECHO - ECHOB
		1206 ; INPUTS:	C - CHARACTER TO ECHO TO CONSOLE
		1207 ;	B - CHARACTER TO BE ECHOED (ECHOB ONLY)
		1208 ; OUTPUTS:	C - CHARACTER ECHOED EXCEPT FOR CARRIAGE
		1209 ; RETURN	RETURN
		1210 ; CALLS:	A,B - CHARACTER ECHOED (ECHOB ONLY)
		1211 ; DESTROYS:	CO
		1212 ; DESCRIPTION:	A,B,F/F'S
		1213 ; REGISTER AND	ECHO TAKES THE CHARACTER INPUT VIA THE C
		1214 ; ARRIAGE RETURN	SENGS THE CHARACTER TO THE CONSOLE. A C
		1215 ; CTER IS ECHOED	IS ECHOED AS A CR/LF AND AN ESCAPE CHARA
		1216 ; TIONS FOR PRO-	AS A \$. ECHOB PROVIDES OUTPUT COMMUNICA
		1217 ; NICATIONS.	GRAMS WHICH USE THE B REGISTER FOR COMMU
		1218 ECHOB:	
F50C C5		1219 PUSH	B : SAVE THE B&C REGISTERS
F50D 48		1220 MOV	C,B
F50E CD14F5		1221 CALL	ECHO : OUTPUT THE CHARACTER
F511 C1		1222 POP	B
F512 78		1223 MOV	A,B : RESTORE REGISTERS AND LEAVE O
		1224 RET	UTPUT DATA IN A
F513 C9		1225 ;	
		1226 ;	
		1227 ;	
		1228 ECHO:	
F514 C5		1229 PUSH	B
F515 41		1230 MOV	B,C
F516 3E1B		1231 MVI	A,ESC
F518 B8		1232 CMP	B
F519 CA25F5		1233 JZ	ECHO4
F51C 3E0D		1234 MVI	A,CR
F51E B8		1235 CMP	B
F51F CA3CF5		1236 JZ	ECHO6
F522 C327F5		1237 JMP	ECHO5
		1238 ECHO4:	
F525 OE24		1239 MVI	C,'\$'
		1240 ECHO5:	
F527 CDDAF4		1241 CALL	CO
F52A 3AF9F8		1242 LDA	LCOUNT
F52D 47		1243 MOV	B,A
F52E 04		1244 INR	B
F52F 3AFAF8		1245 LDA	LLIMIT
F532 B8		1246 CMP	B
F533 DA3CF5		1247 JC	ECHO6
F536 78		1248 MOV	A,B
F537 32F9F8		1249 STA	LCOUNT
F53A C1		1250 POP	B

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ
F53B C9		1251	RET	832	680 304
		1252	ECHO6:		PO 8281
F53C AF		1253	XRA A		8281
F53D 32F9F8		1254	STA LCOUNT		8281
F540 OEOD		1255	MVI C,CR		8281
F542 CDDAF4		1256	CALL CO		8281
F545 OEOA		1257	MVI C,LF		8281
F547 CDDAF4		1258	CALL CO		8281
F54A C1		1259	POP B		8281
F54B C9		1260	RET		8281
		1261	;		8281
		1262	*****		8281
		1263	;		8281
		1264	;		8281
		1265	; FUNCTION: ERROR		8281
		1266	; INPUTS: NONE		8281
		1267	; OUTPUTS: NONE		8281
		1268	; CALLS: ECHO,CROUT,GETCM		8281
		1269	; DESTROYS: A,B,C,F/F'S		8281
		1270	; DESCRIPTION: ERROR PRINTS AN ASTERISK AT THE CONSOLE FOLLOWED		8281
		1271	;		8281
			BY A CARRIAGE RETURN - LINE FEED AND THE		8281
		1272	;		8281
			CONTROL TO THE COMMAND RECOGNIZER.		8281
		1273	;		8281
		1274	ERROR:		8281
F54C OE2A		1275	MVI C,'*' ;		8281
F54E CD14F5		1276	CALL ECHO ;		8281
			SEND THE CHARACTER TO THE CONSOLE		8281
F551 CD06F5		1277	CALL CROUT ;		8281
F554 C380FO		1278	JMP GETCM ;		8281
			GO TO COMMAND RECOGNIZING ROUTINE		8281
		1279	;		8281
		1280	;		8281
		1281	*****		8281
		1282	;		8281
		1283	;		8281
		1284	; FUNCTION: FRET		8281
		1285	; INPUT: NONE		8281
		1286	; OUTPUT: CARRY FLAG ALWAYS 0		8281
		1287	; CALLS: NOTHING		8281
		1288	; DESTROYS: CARRY FLAG		8281
		1289	; DESCRIPTION: FRET IS JUMPED TO BY ANY ROUTINE THAT WISHES TO INDICATE		8281
		1290	A FAILURE ON RETURN. FRET CLEARS THE CARRY FLAG DE-		8281
		1291	NOTING FAILURE THEN RETURNS TO THE CALLING ROUTINE		8281
		1292	;		8281
		1293	;		8281
		1294	FRET:		8281
F557 37		1295	STC ;		8281
F558 3F		1296	CMC ;		8281
			SET CARRY TRUE		8281
			COMPLEMENT CARRY		8281

LOC	OBJ	SEQ	SOURCE STATEMENT
F559 C9		1297	RET
		1298 ;	
		1299 ;	
		1300 ; ****	*****
		1301 ;	
		1302 ;	
		1303 ; FUNCTION: CNVBN	
		1304 ; INPUTS: C - ASCII/HEX CHARACTER	
		1305 ; OUTPUTS: A - 0 TO F HEX (FOUR BITS)	
		1306 ; CALLS: NOTHING	
		1307 ; DESTROYS: A,F/F'S	
		1308 ; DESCRIPTION: CNVBN CONVERTS THE ASCII CHARACTER IN TH E C REGISTER	
		1309 ; TO ITS CORRESPONDING BINARY VALUE. CNVB N DOES NOT CHECK	
		1310 ; THE VALIDITY OF THE INPUT.	
		1311 ;	
		1312 ;	
		1313 CNVBN:	
F55A 79		1314	MOV A,C
F55B D630		1315	SUI '0' ; SUBTRACT ASCII ZERO FROM ARGU MENT
F55D FEOA		1316	CPI 10 ; TEST FOR RESULT = 0 TO 9
F55F F8		1317	RM ; YES - RETURN
F560 D607		1318	SUI 7 ; ELSE RESULT IS BETWEEN 17 AND 23 DECIMAL
F562 C9		1319	RET ; RETURN AFTER SUBTRACTING 7
		1320 ;	
		1321 ;	
		1322 ; ****	*****
		1323 ;	
		1324 ;	
		1325 ; FUNCTION: GETCH	
		1326 ; INPUTS: NONE	
		1327 ; OUTPUTS: C - NEXT CHARACTER FROM INPUT	
		1328 ; CALLS: CI	
		1329 ; DESTROYS: A,C,F/F'S	
		1330 ; DESCRIPTION: GETCH RETURNS THE NEXT CHARACTER IN THE INPUT STREAM TO	
		1331 ; THE CALLING PROGRAM.	
		1332 ;	
		1333 ;	
		1334 GETCH:	
F563 CD7FF4		1335	CALL CI ; GET CHARACTER
F566 E67F		1336	ANI PRTYO ; STRIP PARITY BIT
F568 4F		1337	MOV C,A ; PUT VALUE IN THE C REGISTER A ND RETURN
F569 C9		1338	RET
		1339 ;	
		1340 ;	
		1341 ; ****	*****
		1342 ;	

LOC	OBJ	SEQ	SOURCE STATEMENT	OPCODE	REGS	OPCODE	REGS
		1343 ;					
		1344 ; FUNCTION:	GETHX	038	HL	038	HL
		1345 ; INPUTS:	NONE				
		1346 ; OUTPUTS:	B&C - 16 BIT INTEGER				
		1347 ; ER	D - CHARACTER WHICH TERMINATED THE INTEG				
		1348 ; IMITER	CARRY - 1 IF THE FIRST CHARACTER NOT DEL				
		1349 ; LIMITER	0 IF THE FIRST CHARACTER IS A DE				
		1350 ; CALLS:	GETCH, ECHO, VALDL, VALDG, CNVBN, ERROR				
		1351 ; DESTROYS:	A,B,C,D,E,F/F'S				
		1352 ; DESCRIPTION:	GETHX ACCEPTS A STRING OF HEX DIGITS FRO				
		M THE INPUT AND	RETURNS THERE VALUE AS A 16 BIT BINARY I				
		1353 ; NTEGER. IF	MORE THAN 4 DIGITS ARE ENTERED, ONLY THE				
		1354 ; LAST 4 ARE USED.	LAST 4 ARE USED.				
		1355 ; LIMITER IS EN-	THE NUMBER IS TERMINATED WHEN A VALID DE				
		1356 ; ED AS AN OUTPUT	COUNTERED. THE DELIMITER IS ALSO RETURN				
		1357 ; T HEX DIGITS OR	OF THE FUNCTION. ILLEGAL CHARACTERS (NO				
		1358 ; DELIMITERS) CAUSE AN ERROR OUTPUT.	DELIMITERS) CAUSE AN ERROR OUTPUT.				
		1359 ;					
		1360 ;					
		1361 GETHX:					
F56A E5		1362 PUSH	H ; SAVE H&L				
F56B 210000		1363 LXI	H,O ; CLEAR RESULT REGISTERS				
F56E 1E00		1364 MVI	E,O ; CLEAR DIGIT FLAG				
		1365 GHX05:					
F570 CD63F5		1366 CALL	GETCH ; GET A CHARACTER				
F573 4F		1367 MOV	C,A				
F574 CD14F5		1368 CALL	ECHO ; OUTPUT TO CONSOLE				
F577 CD03F7		1369 CALL	VALDL ; CHECK FOR DELIMITER				
		1370 FALSE	GHX10 ; NO - BRANCH				
F57A D289F5		1371+ JNC	GHX10				
F57D 51		1372 MOV	D,C ; YES - DONE, PUT DELIMITER IN				
		D FOR RETURN					
F57E E5		1373 PUSH	H				
F57F C1		1374 POP	B ; RESULT TO B&C REGISTERS				
F580 E1		1375 POP	H ; RESTORE H&L REGISTERS				
F581 7B		1376 MOV	A,E ; GET FLAG				
F582 B7		1377 ORA	A ; SET F/F'S				
F583 C2BCF6		1378 JNZ	SRET ; IF FLAG NOT ZERO - FLAG FOUND				
		, CY = 1					
F586 CA57F5		1379 JZ	FRET ; FIRST CHARACTER WAS DELIMITER				
		1380 GHX10:					
F589 CDE8F6		1381 CALL	VALDG ; IF NOT DELIMITER CHECK FOR VA				
		LID HEX DIGIT					
		1382 FALSE	ERROR ; ERROR IF NOT VALID DIGIT				
F58C D24CF5		1383+ JNC	ERROR				
F58F CD5AF5		1384 CALL	CNVBN ; ELSE CONVERT TO ITS BINARY VA				
		LUE					

LOC	OBJ	SEQ	SOURCE STATEMENT	DIS	LSD	CSD
F592	1EFF	1385	MVI E, OFFH ; SET DIGIT FLAG NON-0			
F594	29	1386	DAD H ; *2			
F595	29	1387	DAD H ; *4			
F596	29	1388	DAD H ; *8			
F597	29	1389	DAD H ; *16			
F598	0600	1390	MVI B, 0 ; CLEAR UPPER HALF OF B&C PAIR			
F59A	4F	1391	MOV C,A ; BINARY VALUE OF CHARACTER INT			
		0 C				
F59B	09	1392	DAD B ; ADD THIS VALUE TO THE PARTIAL.			
F59C	C370F5	1393	JMP GHX05 ; GET NEXT CHARACTER			
		1394 ;				
		1395 ;				
		1396 ;				
		1397 ; FUNCTION: RSTOR				
		1398 ; INPUTS: NONE				
		1399 ; OUTPUTS: NONE				
		1400 ; CALLS: NOTHING				
		1401 ; DESCRIPTION: RSTOR RESTORES ALL CPU REGISTERS, FLIP-FLIP OPS				
		1402 ; INTERRUPT STATUS, INTERRUPT MASK STACK POSITION				
		1403 ; FROM THEIR RESPECTIVE SAVE LOCATIONS IN MEMORY.				
		1404 ; NOTE - BY RESTORING THE PROGRAM COUNTER,				
		1405 ; EFFECTIVELY TRANSFERS CONTROL OF THE CPU				
		1406 ; TO THE USERS PROGRAM.				
		1407 ;				
		1408 ;				
		1409 ;				
		1410 ;				
		1411 RSTOR:				
F59F	3AE4F8 (13)	1412	LDA ISAV ; GET THE USER INTERRUPT MASK			
F5A2	F618 (7)	1413	ORI 18H ; ENABLE INTERRUPT MASK SETTING			
		AND				
		1414				
		1415 SIM				
F5A4	30 (4)	1416	LDA ISAV ; GET USER INTERRUPT MASK			
F5A5	3AE4F8 (13)	1417	ANI 08H ; SHOULD USER INTERRUPTS BE ENA			
		BLED?				
F5AA	CAB1F5 (7/10)	1418	JZ RSR05 ; NO - LEAVE INTERRUPTS DISABLE			
		D				
F5AD	FB (9)	1419	EI			
F5AE	C3B5F5 (9)	1420	JMP RSR10			
		1421 RSR05:				
F5B1	37 (4)	1422	STC -SET CARRY			
F5B2	D2B5F5 (7/10)	1423	JNC RSR10			
		1424 RSR10:				
F5B5	21DCF8 (10)	1425	LXI H, MNSTK ; SET MONITOR STACK POINTER TO			
		1426				
			START OF MONITOR STACK			
F5B8	F9 (6)	1427	SPHL			
F5B9	D1 (10)	1428	POP D			
F5BA	C1 (10)	1429	POP B			
F5BB	F1 (10)	1430	POP PSW			

LOC	OBJ	SEQ	SOURCE STATEMENT	OPC	LSD	CSD
F5BC	2AE7F8	(16)	1431 LHLD SSAVE ; RESTORE USER STACK POINTER			
F5BF	F9	(6)	1432 SPHL			
F5C0	2AE5F8	(16)	1433 LHLD PSAV ; PUT PROGRAM COUNTER ON TOP OF			
			STACK			
F5C3	E5	(12)	1434 PUSH H			
F5C4	2AE2F8	(16)	1435 LHLD LSAV ; RESTORE H&L REGISTERS			
F5C7	C9	(10)	1436 RET R0 ; JUMP TO USER PROGRAM			
			1437 ;			
			1438 ;			
			1439 ;			
			1440 ;*****			
			1441 ;			
			1442 ;			
			1443 ; FUNCTION: GETNM			
			1444 ; INPUTS: C - COUNT OF NUMBERS TO FIND IN INPUT ST			
			REAM			
			1445 ; OUTPUTS: TOP OF STACK - NUMBERS FOUND IN REVERSE			
			ORDER (LAST ON TOP)			
			1446 ; CALLS: GETHX, HILO, ERROR			
			1447 ; DESTROYS: A,B,C,D,E,H,L,F/F'S			
			1448 ; DESCRIPTION: GETNM FINDS A SPECIFIED COUNT OF NUMBERS			
			(FROM 1 TO 3)			
			1449 ; IN THE INPUT STREAM AND RETURNS THEIR VA			
			LUES ON THE STACK.			
			1450 ; IF TWO OR MORE NUMBERS ARE REQUESTED, TH			
			EN THE FIRST			
			1451 ; MUST BE LESS THAN OR EQUAL TO THE SECOND			
			, OR THEY WILL BE			
			1452 ; SET EQUAL. THE LAST NUMBER MUST BE TERM			
			INATED WITH A			
			1453 ; CARRIAGE RETURN OR AN ERROR INDICATION W			
			ILL RESULT.			
			1454 ;			
			1455 ;			
			1456 GETNM:			
F5C8	2E03		1457 MVI L,3 ; PUT MAXIMUM COUNT INTO L			
F5CA	79		1458 MOV A,C ; GET ACTUAL COUNT			
F5CB	E603		1459 ANI 3 ; FORCE TO MAXIMUM OF 3			
F5CD	C8		1460 RZ ; IF COUNT 0, RETURN			
F5CE	67		1461 MOV H,A ; ELSE, PUT ACTUAL COUNT INTO H			
			1462 GNM05:			
F5CF	CD6AF5		1463 CALL GETHX ; GET A NUMBER FROM INPUT STREA			
			M YAT : FALSE ERROR ; IF NOT THERE - TOO FEW NUMBER			
			S			
F5D2	D24CF5		1465+ JNC ERROR			
F5D5	C5		1466 PUSH B ; ELSE - SAVE NUMBER ON STACK			
F5D6	2D		1467 DCR L ; DECREMENT MAXIMUM ARGUMENT CO			
			UNT			
F5D7	25		1468 DCR H ; DECREMENT ACTUAL ARGUMENT CO			
			NT			
F5D8	CAE4F5		1469 JZ GNM10 ; BRANCH IF NO MORE NUMBERS WAN			
			TED			
F5DB	7A		1470 MOV A,D ; ELSE - GET TERMINATOR TO A			

LOC	OBJ	SEQ	SOURCE STATEMENT	OPCODE	REGISTERS
F5DC	FE0D	1471	CPI CR		; SEE IF CARRIAGE RETURN
F5DE	CA4CF5	1472	JZ ERROR		; IF NOT, IMPROPER TERMINATOR
F5E1	C3CFF5	1473	JMP GNM05		; ELSE, PROCESS NEXT NUMBER
		1474	GNM10:		
F5E4	7A	1475	MOV A,D		; WHEN COUNT=0, CHECK LAST TERM
			INATOR		
F5E5	FE0D	1476	CPI CR		
F5E7	C24CF5	1477	JNZ ERROR		; ERROR IF NOT CARRIAGE RETURN
F5EA	01FFFF	1478	LXI B,0FFFFH		; H&L GETS LARGEST NUMBER
			ER		
F5ED	7D	1479	MOV A,L		; GET WHATS LEFT OF MAX ARGUMENT
			T COUNT		
F5EE	B7	1480	ORA A		; CHECK FOR 0
F5EF	CAF7F5	1481	JZ GNM20		; IF YES, THREE NUMBERS WERE IN
			PUT		
		1482	GNM15:		
F5F2	C5	1483	PUSH B		; IF NOT FILL REMAINING ARGUMENT
			TS WITH 0FFFFH		
F5F3	2D	1484	DCR L		
F5F4	C2F2F5	1485	JNZ GNM15		
		1486	GNM20:		
F5F7	C1	1487	POP B		; GET ARGUMENTS OUT
F5F8	D1	1488	POP D		
F5F9	E1	1489	POP H		
F5FA	CD0DF6	1490	CALL HILO		; SEE IF FIRST >= SECOND
		1491	FALSE		; NO - BRANCH
F5FD	D202F6	1492+	JNC GNM25		
F600	54	1493	MOV D,H		
F601	5D	1494	MOV E,L		; YES - MAKE SECOND EQUAL TO FIRST
			RST		
		1495	GNM25:		
F602	E3	1496	XTHL		; PUT FIRST NO. ON STACK AND GET
			T RETURN ADDRESS		
F603	D5	1497	PUSH D		; PUT SECOND NUMBER ON STACK
F604	C5	1498	PUSH B		; PUT THIRD NUMBER ON STACK
F605	E5	1499	PUSH H		; PUT RETURN ADDRESS ON STACK
		1500	GNM30:		
F606	3D	1501	DCR A		; DECREMENT RESIDUAL COUNT
F607	F8	1502	RM		; IF NEGATIVE, PROPER RESULTS 0
			N STACK		
F608	E1	1503	POP H		; ELSE - GET RETURN ADDRESS
F609	E3	1504	XTHL		; REPLACE TOP RESULT WITH RETURN
			N ADDRESS		
F60A	C306F6	1505	JMP GNM30		; TRY AGAIN
		1506	;		
		1507	;		
		1508	*****		
		1509	;		
		1510	;		
		1511	; FUNCTION: HILO		
		1512	; INPUTS: D&E - 16 BIT INTEGER		
		1513	; H&L - 16 BIT INTEGER		
		1514	; OUTPUTS: CARRY - 0 IF H&L < D&E		
		1515	; 1 IF H&L = D&E		

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ
		1516	; CALLS: NOTHING		
		1517	; DESTROYS: F/F'S		
		1518	; DESCRIPTION: HILO COMPARES THE 2 16 BIT INTEGERS IN H & L AND D&E.		
		1519	; THE CARRY BIT IS SET AS A RESULT OF THE COMPARISON.		
		1520	;		
		1521	;		
		1522	HILO:		
F60D	C5	1523	PUSH B	;	SAVE B&C
F60E	47	1524	MOV B,A	;	SAVE A IN B REGISTER
F60F	E5	1525	PUSH H	;	SAVE H&L
F610	7A	1526	MOV A,D	;	CHECK FOR 0000 IN D&E
F611	B3	1527	ORA E	;	
F612	CA2EF6	1528	JZ HILO5	;	YES - AUTOMATICALLY DONE
F615	23	1529	INX H	;	ELSE - INCREMENT H&L
F616	7C	1530	MOV A,H	;	TEST FOR 0 RESULTS AFTER INCR
			EMENTING		
F617	B5	1531	ORA L	;	
F618	CA2EF6	1532	JZ HILO5	;	YES - H&L MUST HAVE CONTAINED
		FFFF			
F61B	E1	1533	POP H	;	NO - RESTORE ORIGINAL H&L
F61C	D5	1534	PUSH D	;	SAVE D&E
F61D	3EFF	1535	MVI A,OFFH	;	WANT TO TAKE 2'S COMPLEMENT OF
			F D&E		
F61F	AA	1536	XRA D	;	
F620	57	1537	MOV D,A	;	
F621	3EFF	1538	MVI A,OFFH	;	
F623	AB	1539	XRA E	;	
F624	5F	1540	MOV E,A	;	
F625	13	1541	INX D	;	COMPLETES 2'S COMPLEMENT OF D
			&E TO D&E		
F626	7D	1542	MOV A,L	;	
F627	83	1543	ADD E	;	ADD H&L AND D&E
F628	7C	1544	MOV A,H	;	
F629	8A	1545	ADC D	;	THIS OPERATION SETS CARRY PRO
			PERLY		
F62A	D1	1546	POP D	;	RESTORE ORIGINAL D&E CONTENTS
F62B	78	1547	MOV A,B	;	RESTORE ORIGINAL CONTENTS OF
		A			
F62C	C1	1548	POP B	;	RESTORE ORIGINAL CONTENTS OF
		B&C			
F62D	C9	1549	RET	;	
		1550	HILO5:		
F62E	E1	1551	POP H	;	IF H&L = FFFF, THEN CARRY MUS
			T BE SET TO 1		
F62F	78	1552	MOV A,B	;	
F630	C1	1553	POP B	;	RESTORE ORIGINAL CONTENTS OF
		B&C			
F631	C3BCF6	1554	JMP A,SRET	;	SET CARRY AND RETURN
		1555	;		
		1556	;		
		1557	*****	;	*****
		1558	;		

LOC	OBJ	SEQ	SOURCE STATEMENT	936	180	361
		1559	; FUNCTION: DISPC			
		1560	; INPUTS: NONE			
		1561	; OUTPUTS: NONE			
		1562	; CALLS: NMOUT			
		1563	; DESTROYS:			
		1564	; DESCRIPTION: DISPC DISPLAYS THE CURRENT USER PROGRAM COUNTER AT			
		1565	THE CURRENT CONSOLE DEVICE.			
		1566	;			
		1567	;			
		1568	DISPC:			
F634 F5		1569	PUSH PSW			
F635 E5		1570	PUSH H			
F636 21E6F8		1571	LXI H, PSAV+1			
F639 7E		1572	MOV A, M			
F63A CD45F6		1573	CALL NMOUT			
F63D 2B		1574	DCX H			
F63E 7E		1575	MOV A, M			
F63F CD45F6		1576	CALL NMOUT			
F642 E1		1577	POP H			
F643 F1		1578	POP PSW			
F644 C9		1579	RET			
		1580	*****			
		1581	;			
		1582	;			
		1583	;			
		1584	; FUNCTION: NMOUT			
		1585	; INPUTS: A - 8 BIT INTEGER			
		1586	; OUTPUTS: NONE			
		1587	; CALLS: ECHO, PRVAL			
		1588	; DESTROYS: A, B, C, F/F'S			
		1589	; DESCRIPTION: NMOUT CONVERTS THE 8 BIT, UNSIGNED INTEGER IN THE A REG-			
		1590	ISTER INTO TWO ASCII CHARACTERS. THE ASCII CHARACTERS			
		1591	ARE THEN SENT TO THE CONSOLE AT THE CURRENT PRINT POSITION.			
		1592	;			
		1593	;			
		1594	NMOUT:			
F645 E5		1595	PUSH H ; SAVE H&L, DESTROYED BY PRVAL			
F646 F5		1596	PUSH PSW ; SAVE ARGUMENT			
F647 OF		1597	RRC			
F648 OF		1598	RRC			
F649 OF		1599	RRC			
F64A OF		1600	RRC ; GET UPPER 4 BITS TO THE LOWER			
			4 BIT POSITIONS			
F64B E60F		1601	ANI HCHAR ; MASK OUT UPPER 4 BITS			
F64D 4F		1602	MOV C,A			
F64E CD60F6		1603	CALL PRVAL ; CONVERT LOWER 4 BITS TO ASCII			
F651 CD14F5		1604	CALL ECHO ; OUTPUT RESULTS TO CONSOLE			
F654 F1		1605	POP PSW ; GET ARGUMENT BACK			
F655 E60F		1606	ANI HCHAR ; MASK OUT UPPER 4 BITS			
F657 4F		1607	MOV C,A			

LOC	OBJ	SEQ	SOURCE STATEMENT	BBB	BBB	BBB
F658	CD60F6	1608	CALL PRVAL ; CONVERT LOWER 4 BITS TO ASCII			
F65B	CD14F5	1609	CALL ECHO ; OUTPUT RESULTS TO CONSOLE			
F65E	E1	1610	POP H ; RESTORE H&L			
F65F	C9	1611	RET			
		1612 ;				
		1613 ;				
		1614 ;*****				
		1615 ;				
		1616 ;				
		1617 ; FUNCTION: PRVAL				
		1618 ; INPUTS: C - INTEGER (RANGE 0 - F)				
		1619 ; OUTPUTS: C - ASCII CHARACTER				
		1620 ; CALLS: NOTHING				
		1621 ; DESTROYS: B,C,H,L,F/F'S				
		1622 ; DESCRIPTION: PRVAL CONVERTS A NUMBER IN THE RANGE 0 T O F HEX TO THE				
		1623 ; CORRESPONDING ASCII CHARACTER, 0-9, A-F.				
		PRVAL DOES NOT				
		1624 ; CHECK THE VALIDITY OF THE INPUT ARGUMENT				
		1625 ;				
		1626 ;				
		1627 PRVAL:				
F660	21BDF7	1628	LXI H,DIGTB ; ADDRESS OF LOOK-UP TABLE			
F663	0600	1629	MVI B,0 ; CLEAR UPPER 8 BITS OF B&C			
F665	09	1630	DAD B ; ADD DIGIT VALUE TO ADDRESS			
F666	4E	1631	MOV C,M ; FETCH CHARACTER FROM MEMORY			
F667	C9	1632	RET			
		1633 ;				
		1634 ;				
		1635 ;*****				
		1636 ;				
		1637 ;				
		1638 ; FUNCTION: REGDS				
		1639 ; INPUTS: NONE				
		1640 ; OUTPUTS: NONE				
		1641 ; CALLS: ECHO, NMOUT, ERROR, CROUT				
		1642 ; DESTROYS: A,B,C,D,E,H,L,F/F'S				
		1643 ; DESCRIPTION: REGDS DISPLAYS THE CONTENTS OF THE REGIS TER SAVE LOCATIONS,				
		1644 ; IN FORMATTED FORM, TO THE CONSOLE. THE DISPLAY IS DRIVEN				
		1645 ; FROM A TABLE, RTAB, WHICH CONTAINS THE R EGISTERS PRINT				
		1646 ; SYMBOL, SAVE LOCATION ADDRESS, AND LENGTH H (8 OR 16 BITS)				
		1647 ;				
		1648 ;				
		1649 REGDS:				
F668	21CDF7	1650	LXI H,RTAB ; LOAD H&L WITH THE STARTING AD DRESS OF TABLE			
		1651 REGDS:				
F66B	4E	1652	MOV C,M ; GET PRINT SYMBOL OF REGISTER			

LOC	OBJ	SEQ	SOURCE STATEMENT		
F66C	79	1653	MOV A,C		
F66D	B7	1654	ORA A	; TEST FOR 0 - END OF TABLE	
F66E	C275F6	1655	JNZ REC10	; IF NOT END, BRANCH	
F671	CD06F5	1656	CALL CROUT	; ELSE - SEND CR/LF TO CONSOLE	
			TO TERMINATE DISPLAY		
F674	C9	1657	RET		
		1658	REG10:		
F675	CD14F5	1659	CALL ECHO	; OUTPUT IDENTIFIER	
F678	OE3D	1660	MVI C,'='		
F67A	CD14F5	1661	CALL ECHO	; OUTPUT EQUALS SIGN	
F67D	23	1662	INX H	; POINT TO START OF SAVE LOC AD	
			DRESS		
F67E	5E	1663	MOV E,M	; GET LOWER 8 BITS OF SAVE LOCA	
F67F	16F8	1664	MVI D, RAMST SHR 8	; PUT UPPER BYTE OF ADD	
			RES IN SAVE LOC		
F681	23	1665	INX H	; POINT TO FLAG LENGTH	
F682	1A	1666	LDAX D	; GET CONTENTS OF SAVE ADDRESS	
F683	CD45F6	1667	CALL NMOUT	; DISPLAY ON CONSOLE	
F686	7E	1668	MOV A,M	; GET LENGTH FLAG	
F687	B7	1669	ORA A	; SET SIGN F/F	
F688	CA90F6	1670	JZ REG15	; IF 0, REGISTER IS 8 BITS	
F68B	1B	1671	DCX D	; ELSE - MOVE POINTER TO REMAIN	
			DER OF REGISTER		
F68C	1A	1672	LDAX D	; GET LOWER 8 BITS	
F68D	CD45F6	1673	CALL NMOUT	; DISPLAY LOWER 8 BITS	
		1674	REG15:		
F690	OE20	1675	MVI C,' '		
F692	CD14F5	1676	CALL ECHO	; USE SPACE AS SEPARATOR	
F695	23	1677	INX H	; POINT TO START OF NEXT TABLE	
			ENTRY		
F696	C36BF6	1678	JMP REG05	; DO NEXT REGISTER	
		1679 ;			
		1680 ;			
		1681 ; FUNCTION: TERROR			
		1682 ; DESCRIPTION: CALLED BY THE LOAD ROUTINE TO INDICATE A			
		N ERROR.			
		1683 ;			
		1684 TERROR:			
F699	3E40	1685	MVI A,40H		
F69B	DSFO	1686	OUT PORTA		
F69D	218CF7	1687	LXI H,TEMSG		
F6A0	060E	1688	MVI B,TEMSGL		
F6A2	C377F0	1689	JMP MSGL		
		1690 ;			
		1691 ;			
		1692 ;			
		1693 ;*****			
		*****			
		1694 ;			
		1695 ;			
		1696 ; FUNCTION: RGADR			
		1697 ; INPUTS: C - REGISTER DENOTING CHARACTER			
		1698 ; OUTPUTS: B&C - ADDRESS OF ENTRY IN RTAB (REGISTER			
		TABLE)			

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ
		1699 ;	CALLS: ERROR	03B	03D
		1700 ;	DESTROYS: A,B,C,D,E,H,L,F/F'S	03B	03D
		1701 ;	DESCRIPTION: RGADR TAKES A SINGLE CHARACTER AS INPUT.		
			THIS CHARACTER		
		1702 ;	DENOTES A REGISTER. RGADR SEARCHES THE		
			REGISTER TABLE FOR		
		1703 ;	A MATCH TO THE INPUT. IF ONE OCCURS, RG		
			ADR RETURNS THE		
		1704 ;	ADDRESS OF THE SAVE LOCATION CORRESPONDING		
			TO THE REGISTER.		
		1705 ;	IF NO MATCH OCCURS, THEN THE REGISTER IDENTIFIER IS ILLEGAL		
		1706 ;	AND CONTROL IS PASSED TO THE ERROR HANDLER.		
		1707 ;			
		1708 ;			
		1709 RGADR:			
F6A5 21CDF7		1710 LXI H,RTAB ;	H&L GETS ADDRESS OF TABLE STA		
		RT			
F6A8 110300		1711 LXI D,RTABS ;	D&E GET THE SIZE OF THE TABLE		
		ENTRY			
		1712 RGA05:			
F6AB 7E		1713 MOV A,M ;	GET REGISTER IDENTIFIER		
F6AC B7		1714 ORA A ;	CHECK FOR END OF TABLE (IDENT		
		IFIER = 0)			
F6AD CA4CF5		1715 JZ ERROR ;	IF AT END OF TABLE, ARGUMENT		
		WAS ILLEGAL			
F6B0 B9		1716 CMP C ;	COMPARE TABLE ENTRY AND ARGUM		
		ENT			
F6B1 CABBF6		1717 JZ RGA10 ;	IF EQUAL, IDENTIFIER WAS FOUN		
		D			
F6B4 19		1718 DAD D ;	ELSE - INCREMENT TABLE POINTE		
		R TO NEXT ENTRY			
F6B5 C3ABF6		1719 JMP RGA05 ;	TEST NEXT ENTRY		
		1720 RGA10:			
F6B8 23		1721 INX H ;	IF A MATCH, INCREMENT POINTER		
		TO SAVE LOC ADDRESS			
F6B9 44		1722 MOV B,H ;	AND GET VALUE TO B&C FOR RETU		
		RN			
F6BA 4D		1723 MOV C,L ;			
F6BB C9		1724 RET			
		1725 ;			
		1726 ;			
		1727 ;*****			
		1728 ;			
		1729 ;			
		1730 ; FUNCTION: SRET			
		1731 ; INPUT: NONE			
		1732 ; OUTPUT: CARRY=1			
		1733 ; CALLS: NOTHING			
		1734 ; DESTROYS: CARRY			
		1735 ; DESCRIPTION: SRET IS JUMPED TO BY ROUTINES WISHING TO			
		RETURN SUCCESS.			
		1736 ;	SRET SETS THE CARRY TRUE AND RETURNS TO		

LOC	OBJ	SEQ	SOURCE STATEMENT
		SEQ	THE CALLER OF THE
		1737 ;	ROUTINE INVOKING SRET (NEXT VECTOR ON ST
		1738 ;	ACK).
		1739 ;	
		1740 SRET:	
F6BC 37		1741 STC	; SET THE CARRY TRUE
F6BD C9		1742 RET	
		1743 ;	
		1744 ;	
		1745 ;*****	*****
		1746 ;	
		1747 ;	
		1748 ; FUNCTION: STHFO	
		1749 ; INPUTS: D&E - 16 BIT ADDRESS OF BYTE TO BE STORE	
		1750 ; D INTO	
		1751 ; OUTPUTS: NONE	
		1752 ; CALLS: STHLF	
		1753 ; DESTROYS: A,B,C,H,L,F/F'S	
		1754 ; DESCRIPTION: STHFO CHECKS THE HALF BYTE FLAG IN TEMP	
		1755 ; TO SEE IF IT IS	
		1756 ; SET TO LOWER. IF SO, STHFO STORES A O T	
		1757 ; O PAD OUT THE LOWER	
		1758 ; HALF OF THE ADDRESSED BYTE. OTHERWISE,	
		1759 ; THE ROUTINE TAKES	
		1760 LDA TEMP	; GET HALF BYTE FLAG
F6BE 3AF4FB		1761 ORA A	; SET F/F'S
F6C1 B7		1762 RNZ	; IF SET TO UPPER DONT DO ANYTH
F6C2 C0		ING	
F6C3 0E00		1763 MVI C,O	; ELSE - STORE THE VALUE O
F6C5 CDC9F6		1764 CALL STHLF	; STORES THE O
F6C8 C9		1765 RET	
		1766 ;	
		1767 ;	
		1768 ;*****	*****
		1769 ;	
		1770 ;	
		1771 ; FUNCTION: STHLF	
		1772 ; INPUTS: C - 4 BIT VALUE TO BE STORED IN HALF-BYT	
		1773 ; E	
		1774 ; D&E - 16 BIT ADDRESS OF BYTE TO BE STORE	
		1775 ; D INTO	
		1776 ; OUTPUTS: NONE	
		1777 ; CALLS: NOTHING	
		1778 ; DESTROYS: A,B,C,H,L,F/F'S	
		1779 ; DESCRIPTION: STHLF TAKES THE 4 BIT VALUE IN C AND STO	
		1780 ; RES IT IN HALF OF	
		1781 ; THE BYTE ADDRESSED BY THE D&E REGISTERS.	
		1782 ; THE HALF BYTE	

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ
		1779 ;	USED (EITHER UPPER OR LOWER) IS DENOTED		
		BY THE VALUE OF			
		1780 ;	THE FLAG IN TEMP. STHLF ASSUMES THAT TH		
		E FLAG HAS BEEN			
		1781 ;	PREVIOUSLY SET TO THE DESIRED VALUE.		
		1782 ;			
		1783 ;			
		1784 STHLF:			
F6C9 D5		1785	PUSH D		
F6CA E1		1786	POP H ; MOVE ADDRESS TO H&L		
F6CB 79		1787	MOV A,C ; GET VALUE		
F6CC E60F		1788	ANI OFH ; FORCE TO 4 BIT LENGTH		
F6CE 4F		1789	MOV C,A ; PUT VALUE BACK		
F6CF 3AF4F8		1790	LDA TEMP ; GET HALF-BYTE FLAG		
F6D2 B7		1791	ORA A ; CHECK FOR LOWER HALF		
F6D3 C2DCF6		1792	JNZ STH05 ; BRANCH IF NOT		
F6D6 7E		1793	MOV A,M ; ELSE - GET BYTE		
F6D7 E60F		1794	ANI OFOH ; CLEAR LOWER 4 BITS		
F6D9 B1		1795	ORA C ; GET VALUE		
F6DA 77		1796	MOV M,A ; PUT IT BACK		
F6DB C9		1797	RET		
		1798 STH05:			
F6DC 7E		1799	MOV A,M ; IF UPPER HALF, GET BYTE		
F6DD E60F		1800	ANI OFH ; CLEAR UPPER 4 BITS		
F6DF 47		1801	MOV B,A ; SAVE BYTE IN B		
F6E0 79		1802	MOV A,C ; GET VALUE		
F6E1 0F		1803	RRC		
F6E2 0F		1804	RRC		
F6E3 0F		1805	RRC		
F6E4 0F		1806	RRC ; MOVE LOWER 4 BITS INTO UPPER		
		4 BITS			
F6E5 B0		1807	ORA B ; OR IN VALUE		
F6E6 77		1808	MOV M,A ; PUT IT BACK		
F6E7 C9		1809	RET		
		1810 ;			
		1811 ;			
		1812 ;*****			
		1813 ;			
		1814 ;			
		1815 ; FUNCTION: VALDG			
		1816 ; INPUTS: C - ASCII CHARACTER			
		1817 ; OUTPUTS: CARRY - 1 IF CHARACTER REPRESENTS VALID			
		HEX DIGIT			
		1818 ;	0 IF OTHERWISE		
		1819 ; CALLS: NOTHING			
		1820 ; DESTROYS: A, F/F'S			
		1821 ; DESCRIPTION: VALDG RETURNS SUCCESS IF ITS INPUT ARGUM			
		ENT IS AN ASCII			
		1822 ;	CHARACTER REPRESENTING A VALID HEX DIGIT		
		(0-9, A-F), AND			
		1823 ;	A FAILURE OTHERWISE.		
		1824 ;			
		1825 ;			
		1826 VALDG:			

LOC	OBJ	SEQ	SOURCE STATEMENT			
F6E8	79	1827	MOV	A,C		
F6E9	FE30	1828	CPI	'0'	; TEST FOR 0	
F6EB	FA57F5	1829	JM	FRET	; IF CODE IS LESS, CANNOT BE VA	
F6EE	FE39	1830	LID	CPI	'9'	; ELSE - SEE IF CODE IS IN RANG
			E 0 - 9			
F6F0	FAECF6	1831	JM	SRET	; CODE BETWEEN 0 AND 9	
F6F3	CABCF6	1832	JZ	SRET	; CODE EQUALS 9	
F6F6	FE41	1833	CPI	'A'	; COMPARE WITH LETTER A	
F6F8	FA57F5	1834	JM	FRET	; CANNOT BE LESS THAN A AND BE	
			VALID NOW			
F6FB	FE47	1835	CPI	'G'	; COMPARE TO LETTER G	
F6FD	F257F5	1836	JP	FRET	; NO - GREATER THAN F	
F700	C3BCF6	1837	JMP	SRET	; CODE IS FROM A TO F INCLUSIVE	
		1838	;			
		1839	;			
		1840	;*****			
		1841	;			
		1842	;			
		1843	; FUNCTION:	VALDL		
		1844	; INPUTS:	C - CHARACTER		
		1845	; OUTPUTS:	CARRY - 1 IF THE INPUT ARGUMENT IS A VAL		
		1846	ID DELIMITER	O OTHERWISE		
		1847	CALLS:	NOTHING		
		1848	DESTROYS:	A, F/F'S		
		1849	DESCRIPTION:	VALDL RETURNS SUCCESS IF ITS INPUT ARGUM		
		1850	ENT IS A VALID	DELIMITER CHARACTER (SPACE, COMMA, CARRIAG		
		1851	E RETURN) AND	E RETURN)		
		1852	1853	FAILURE OTHERWISE.		
		1854	VALDL:			
F703	79	1855	MOV	A,C		
F704	FE2C	1856	CPI	','	; CHECK FOR COMMA	
F706	CABCF6	1857	JZ	SRET		
F709	FE0D	1858	CPI	ODH	; CHECK FOR CARRIAGE RETURN	
F70B	CABCF6	1859	JZ	SRET		
F70E	FE20	1860	CPI	' '	; CHECK FOR SPACE	
F710	CABCF6	1861	JZ	SRET		
F713	C357F5	1862	JMP	FRET	; ERROR IF NONE OF THE ABOVE	
		1863	;			
		1864	;			
		1865	;			
		1866	;			
		1867	WAIT:			
F716	C5	1868	PUSH	B		
F717	E5	1869	PUSH	H		
F718	060C	1870	MVI	B,12		
F71A	210000	1871	LXI	H,O		
		1872	WAIT1:			
F71D	2D	1873	DCR	L	; EACH SUB LOOP ~ 1200US	
F71E	C21DF7	1874	JNZ	WAIT1		

LOC	OBJ	SEQ	SOURCE STATEMENT	936	180	304
F721	25	1875	DCR H ; TOTAL LOOP ~ 0.3SEC			
F722	C21DF7	1876	JNZ WAIT1			
F725	05	1877	DCR B			
F726	C21DF7	1878	JNZ WAIT1 ; DO IT TWELVE TIMES			
F729	E1	1879	POP H			
F72A	C1	1880	POP B			
F72B	C9	1881	RET			
		1882 ;				
		1883 ;				
		1884 ;				
		1885 LIMITST:				
F72C	DBFO	1886	IN PORTA ; READ STATUS PORT			
F72E	E602	1887	ANI 02H ; CHECK LINE LIMIT JUMPER (0=IN =32CHAR/LINE)			
F730	C247FO	1888	JNZ SIORET ; LEAVE LLIMIT=63			
F733	3E1F	1889	MVI A,31			
F735	32FAF8	1890	STA LLIMIT ; OTHERWISE STORE 31			
F738	3E01	1891	MVI A,01			
F73A	D3F8	1892	OUT CSR ; INITIALIZE 8155			
F73C	32F8F8	1893	STA USCSR			
F73F	C347FO	1894	JMP SIORET			
		1895 ;				
		1896 ;				
		1897 ;				
		1898 ;				
		1899 ;				
		1900 ;*****				
		1901 ;*****				
		1902 ;				
		1903 ;	MONITOR TABLES			
		1904 ;				
		1905 ;*****				
		1906 ;				
		1907 ;				
		1908 SINON: ; SIGNON MESSAGE				
F742	0D	1909	DB CR,LF,'EXPLORER-85 VER 1.4',CR,LF			
F743	0A					
F744	4558504C					
F748	4F524552					
F74C	2D383520					
F750	20205645					
F754	5220312E					
F758	34					
F759	0D					
F75A	0A					
F75B	434F5059	1910	DB 'COPYRIGHT 1979',CR,LF			
F75F	52494748					
F763	54203139					
F767	3739					
F769	0D					
F76A	0A					
F76B	4E455452	1911	DB 'NETRONICS R&D',CR,LF			

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	OBJ	OBJ
F76F	4F4E4943			4943	4943	4943
F773	53205226			53205226	53205226	53205226
F777	44			44	44	44
F778	OD			OD	OD	OD
F779	OA			OA	OA	OA
F77A	4E455720	1912	DB 'NEW MILFORD, CT.',CR,LF	4E455720	4E455720	4E455720
F77E	4D494C46			4D494C46	4D494C46	4D494C46
F782	4F52442C			4F52442C	4F52442C	4F52442C
F786	2043542E			2043542E	2043542E	2043542E
F78A	OD			OD	OD	OD
F78B	OA			OA	OA	OA
004A		1913	LSGNON EQU \$-SINON ; LENGTH OF SIGNON MESSAGE	004A	004A	004A
		1914	;			
		1915	;			
		1916	TEMSG:			
F78C	OD	1917	DB CR,LF,'TAPE ERROR',CR,LF	F78C	OD	OD
F78D	OA			F78D	OA	OA
F78E	54415045			F78E	54415045	54415045
F792	20455252			F792	20455252	20455252
F796	4F52			F796	4F52	4F52
F798	OD			F798	OD	OD
F799	OA			F799	OA	OA
000E		1918	TEMSGL EQU \$-TEMSG ; LENGTH OF TAPE ERROR MESSAGE	000E	000E	000E
		1919	;			
		1920	;			
		1921	CADR: ROUTINES ; TABLE OF ADDRESSES OF COMMAND			
F79A	0000	1922	DW O ; DUMMY	F79A	0000	0000
F79C	6CF2	1923	DW XCMD	F79C	6CF2	6CF2
F79E	C6F1	1924	DW SCMD	F79E	C6F1	C6F1
F7A0	A2F1	1925	DW CCMD	F7A0	A2F1	A2F1
F7A2	F4FO	1926	DW TCMD	F7A2	F4FO	F4FO
F7A4	82F1	1927	DW MCMD	F7A4	82F1	82F1
F7A6	1AF1	1928	DW GCMD	F7A6	1AF1	1AF1
F7A8	38F1	1929	DW ICMD	F7A8	38F1	38F1
F7AA	B0FO	1930	DW DCMD	F7AA	B0FO	B0FO
F7AC	6BF4	1931	DW FCMD	F7AC	6BF4	6BF4
F7AE	E5F2	1932	DW RCMD	F7AE	E5F2	E5F2
F7B0	93F3	1933	DW LCMD	F7B0	93F3	93F3
		1934	;			
		1935	CTAB: TERS ; TABLE OF VALID COMMAND CHARAC			
F7B2	4C	1936	DB 'L'	F7B2	4C	4C
F7B3	52	1937	DB 'R'	F7B3	52	52
F7B4	46	1938	DB 'F'	F7B4	46	46
F7B5	44	1939	DB 'D'	F7B5	44	44
F7B6	49	1940	DB 'I'	F7B6	49	49
F7B7	47	1941	DB 'G'	F7B7	47	47
F7B8	4D	1942	DB 'M'	F7B8	4D	4D
F7B9	54	1943	DB 'T'	F7B9	54	54
F7BA	43	1944	DB 'C'	F7BA	43	43
F7BB	53	1945	DB 'S'	F7BB	53	53
F7BC	58	1946	DB 'X'	F7BC	58	58
000B		1947	NCMDS EQU \$-CTAB ; NUMBER OF VALID COMMANDS	000B	000B	000B
		1948	;			

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ	190	304
		1949 ;		0005	00 3153	
		1950 ;		0006	00 3154	
		1951 DIGTB:		0007	00 3155	
F7BD 30		1952	DB '0'	0008	00 3156	
F7BE 31		1953	DB '1'	0009	00 3157	
F7BF 32		1954	DB '2'	0010	00 3158	
F7C0 33		1955	DB '3'	0011	00 3159	
F7C1 34		1956	DB '4'	0012	00 3160	
F7C2 35		1957	DB '5'	0013	00 3161	
F7C3 36		1958	DB '6'	0014	00 3162	
F7C4 37		1959	DB '7'	0015	00 3163	
F7C5 38		1960	DB '8'	0016	00 3164	
F7C6 39		1961	DB '9'	0017	00 3165	
F7C7 41		1962	DB 'A'	0018	00 3166	
F7C8 42		1963	DB 'B'	0019	00 3167	
F7C9 43		1964	DB 'C'	0020	00 3168	
F7CA 44		1965	DB 'D'	0021	00 3169	
F7CB 45		1966	DB 'E'	0022	00 3170	
F7CC 46		1967	DB 'F'	0023	00 3171	
		1968 ;		0024	00 3043	
		1969 ;		0025	00 3044	
		1970 RTAB:		0026	00 3045	
F7CD 41		1971	DB 'A'	0027	00 3046	
F7CE E1		1972	DB ASAV AND OFFH	0028	00 3047	
			; ADDRESS OF REGISTER S	0029	00 3048	
F7CF 00		1973	DB 0	0030	00 3049	
0003		1974 RTABS EQU \$-RTAB	; LENGTH OF REGISTER	0031	00 3050	
		E	; SIZE OF AN ENTRY IN THIS TABL	0032	00 3051	
F7D0 42		1975	DB 'B'	0033	00 3052	
F7D1 DF		1976	DB BSAV AND OFFH	0034	00 3053	
F7D2 00		1977	DB 0	0035	00 3054	
F7D3 43		1978	DB 'C'	0036	00 3055	
F7D4 DE		1979	DB CSAV AND OFFH	0037	00 3056	
F7D5 00		1980	DB 0	0038	00 3057	
F7D6 44		1981	DB 'D'	0039	00 3058	
F7D7 DD		1982	DB DSAV AND OFFH	0040	00 3059	
F7D8 00		1983	DB 0	0041	00 3060	
F7D9 45		1984	DB 'E'	0042	00 3061	
F7DA DC		1985	DB ESAV AND OFFH	0043	00 3062	
F7DB 00		1986	DB 0	0044	00 3063	
F7DC 46		1987	DB 'F'	0045	00 3064	
F7DD EO		1988	DB FSAV AND OFFH	0046	00 3065	
F7DE 00		1989	DB 0	0047	00 3066	
F7DF 49		1990	DB 'I'	0048	00 3067	
F7EO E4		1991	DB ISAV AND OFFH	0049	00 3068	
F7E1 00		1992	DB 0	0050	00 3069	
F7E2 48		1993	DB 'H'	0051	00 3070	
F7E3 E3		1994	DB HSAV AND OFFH	0052	00 3071	
F7E4 00		1995	DB 0	0053	00 3072	
F7E5 4C		1996	DB 'L'	0054	00 3073	
F7E6 E2		1997	DB LSAV AND OFFH	0055	00 3074	
F7E7 00		1998	DB 0	0056	00 3075	
F7E8 53		1999	DB 'S'	0057	00 3076	
F7E9 E8		2000	DB SSAV+1 AND OFFH	0058	00 3077	
F7EA 01		2001	DB 1	0059	00 3078	

LOC	OBJ	SEQ	SOURCE STATEMENT	OBJ
		2002	DB 'P'	
F7EB	50	2003	DB PSAV+1 AND OFFH	
F7EC	E6	2004	DB 1	
F7ED	01	2005	DB 0 ; END OF TABLE MARKERS	
F7EE	00	2006	DB 0	
F7EF	00	2007 ;		
		2008 ;		
		2009 ;*****;		
		2010 ;		
		2011 ;	SPACE RESERVED FOR THE MONITOR S	
		2012 TACK		
		2013 ;*****;		
		2014 ;		
F8DC		2015 ORG MNSTK ; START OF MONITOR STACK		
		2016 ;		
		2017 ;	SAVE LOCATIONS FOR USER REGISTERS	
		2018 ;		
F8DC	00	2019 ESAVE: DB 0		
F8DD	00	2020 DSAVE: DB 0		
F8DE	00	2021 CSAV: DB 0		
F8DF	00	2022 BSAV: DB 0		
F8E0	00	2023 FSAV: DB 0		
F8E1	00	2024 ASAVER: DB 0		
F8E2	00	2025 LSAVER: DB 0		
F8E3	00	2026 HSAVER: DB 0		
F8E4	00	2027 ISAV: DB 0		
		2028 PSAV:		
F8E5	00	2029 PCLSV: DB 0		
F8E6	00	2030 PCHSV: DB 0		
		2031 SSAV:		
F8E7	00	2032 SPLSV: DB 0		
F8E8	00	2033 SPHSV: DB 0		
		2034 ;		
		2035 ;*****;		
		2036 ;		
		2037 ;	MONITOR STORAGE LOCATIONS	
		2038 ;		
		2039 ;*****;		
		2040 ;		
		2041 ;		
F8E9	0000	2042 BITTIME: DW 0		
F8EB	0000	2043 HALFBIT: DW 0		
F8ED	0000	2044 CURAD: DW 0		
F8EF	00	2045 CURDT: DB 0		
0004		2046 OBUFF: DS 4		
		2047 TEMP:		
F8F4	00	2048 RGPTR: DB 0		
F8F5	00	2049 PROGNO: DB 0		
F8F6	00	2050 IBUFF: DB 0		
F8F7	00	2051 NEWLN: DB 0		
		2001		

LOC	OBJ	SEQ	SOURCE STATEMENT	REGS	REGS	REGS	REGS
F8F8 00		2052	USCSR: DB 0	2884 A F8A9	R0TR	2889 A F8A9	R0ADH
F8F9 00		2053	LCOUNT: DB 0	2885 A F8B0	R0TR	2886 A F8B0	R0ADH
F8FA 00		2054	LLIMIT: DB 0	2887 A F8B1	STORET A F042	2888 A F8B1	SHTR
F8FB 0000		2055	DIN: DW 0	2888 A F8B2	STOREC A F04C	2889 A F8B2	SHTR
F8FD 0000		2056	DOUT: DW 0	2889 A F8B3	TEET A F04D	288A A F8B3	SHTR
		2057	;	288A A F8B4	START A F05E	288B A F8B4	SHTR
		2058	END	288B A F8B5	SPTR A F05F	288C A F8B5	SHTR
<b>PUBLIC SYMBOLS</b>							
<b>EXTERNAL SYMBOLS</b>							
<b>USER SYMBOLS</b>							

ASAV A F8E1	BC0 A F4D2	BIT05 A F382	BIT10 A F372
BIT15 A F37B	BIT20 A F384	BIT25 A F38D	BIT30 A F391
BITIN A F3E9	BITIN1 A F3EA	BITIN2 A F3F1	BITIN4 A F3FE
BITO A F369	BITSI A 0009	BITSO A 000B	BITTIM A F8E9
BOOT A F000	BRCHR A 001B	BRI1 A F441	BRI3 A F449
BRI4 A F44B	BRID A F43C	BRTAB A F7FA	BSAV A F8DF
BYTO1 A F40E	BYTO2 A F415	BYTO3 A F42B	BYTO4 A F431
BYTIN A F407	BYTO1 A F34E	BYTO2 A F362	BYTO3 A F363
BYTOUT A F34A	CADR A F79A	CCM05 A F1A7	CCM10 A F1B2
CCM15 A F1C2	CCMD A F1A2	CI A F47F	CI1 A F48E
CI2 A F496	CI3 A F49E	CI4 A F4A1	CI5 A F4B6
CIN A F48A	CNVBN A F55A	CO A F4DA	CO1 A F4EA
CO2 A F4F2	COUT A F4E5	CR A 000D	CROUT A F506
CSAV A F8DE	CSR A 00F8	CTAB A F7B2	CURAD A F8ED
CURDT A F8EF	DATA A F328	DCM05 A F0C9	DCM06 A F0C2
DCM07 A F0C4	DCM10 A F0D4	DCM15 A FOE9	DCMD A FOBO
DIGTB A F7BD	DIN A F8FB	DISPC A F634	DOUT A F8FD
DSAV A F8DD	ECHO4 A F525	ECHO5 A F527	ECHO6 A F53C
ECHO A F514	ECHOB A F50C	ERROR A F54C	ESAV A F8DC
ESC A 001B	FALSE + 0001	FCM05 A F474	FCM10 A F475
FCMD A F46B	FRET A F557	FSAV A F8E0	GCM05 A F12F
GCM10 A F135	GCMD A F11A	GETCH A F563	GETCM A F080
GETHX A F56A	GETNM A F5C8	GHX05 A F570	GHX10 A F589
GNM05 A F5CF	GNM10 A F5E4	GNM15 A F5F2	GNM20 A F5F7
GNM25 A F602	GNM30 A F606	GTC05 A F09A	GTC10 A F0A6
GTC03 A F08D	HALFB1 A F8EB	HCHAR A 000F	HIL05 A F62E
HILD A F60D	HSAV A F8E3	IBUFF A F8F6	ICM05 A F143
ICM10 A F16B	ICM20 A F173	ICM25 A F179	ICMD A F138
INVRT A 0OFF	ISAV A F8E4	LCMD A F393	LCMD1 A F3A4
LCMD2 A F3A6	LCMD3 A F3AF	LCMD4 A F3B3	LCMD6 A F3D3
LCOUNT A F8F9	LDR1 A F301	LF A 000A	LIMTST A F72C
LLIMIT A F8FA	LOWER A 0000	LSAV A F8E2	LSGNON A 004A
MCM05 A F18A	MCMD A F182	MESSAGE A F072	MNSTK A F8DC
MSGL A F077	NCMDS A 000B	NEWLN A F8F7	NMOUT A F645
OBUFF A F8F0	PCHSV A F8E6	PCLSV A F8E5	PIN A F4BB
PORTA A 00F0	PORTAD A 00F2	PORTB A 00F1	PORTBD A 00F3
PORTC A 00FB	POUT A F4C6	PRLIN A 00FA	PRLOUT A 00F9
PROGNO A F8F5	PRTYD A 007F	PRVAL A F660	PSAV A F8E5
RAMST A F800	RCMD A F2E5	REG05 A F66B	REG10 A F675
REG15 A F690	REGDS A F668	RGA05 A F6AB	RGA10 A F6B8

RGADR	A F6A5	RGPTR	A F8F4	RMUSE	A 0022	RSAVE	A F04B
RSR05	A F5B1	RSR10	A F5B5	RST55	A F8A6	RST65	A F8A3
RSTOR	A F59F	RTAB	A F7CD	RTABS	A 0003	SCMD	A F1C6
SINON	A F742	SIORET	A F047	SIOTST	A F435	SPHSV	A F8E8
SPLSV	A F8E7	SRET	A F6BC	SSAV	A F8E7	SSTRT	A 0080
START	A F003	START1	A F03F	STH05	A F6DC	STHFO	A F6BE
STHLF	A F6C9	STOPB	A 0040	STP05	A F1DE	STP15	A F1EA
STP15A	A F1FO	STP16	A F201	STP20	A F207	STP21	A F21C
STP22	A F223	STP23	A F226	STP25	A F238	STRT	A 00C0
TCM00	A F10E	TCM01	A F114	TCMD	A F0F4	TEMP	A F8F4
TEMSG	A F78C	TEMMSG	A 000E	TERM	A 001B	TERROR	A F699
TIMHI	A 00FD	TIMLO	A 00FC	TRAIL	A F339	TRUE	+ 0000
UNMSK	A 000E	UPPER	A 00FF	USCSR	A F8FB	USINT	A F8A0
VALDG	A F6E8	VALDL	A F703	WAIT	A F716	WAIT1	A F71D
XCM05	A F27C	XCM10	A F28B	XCM15	A F298	XCM18	A F2A3
XCM20	A F2BC	XCM25	A F2D3	XCM27	A F2D4	XCM30	A F2DC
XCMD	A F26C						

ASSEMBLY COMPLETE; NO ERRORS